

WEST**End of Result Set**

Generate Collection

Print

L1: Entry 16 of 16

File: USPT

Jul 29, 1997

DOCUMENT-IDENTIFIER: US 5652751 A

TITLE: Architecture for mobile radio networks with dynamically changing topology using virtual subnets

Detailed Description Text (99):

In summary, the present architecture comprises a logical topology of physical and virtual subnets, and corresponding addressing, mobility management and routing schemes. The architecture is especially applicable to mobile radio networks and accommodates dynamic topology changes due to relative movement of network nodes. The architecture partitions a mobile radio network into logically independent subnetworks. Network nodes are members of physical and virtual subnets and may change their affiliations with these subnets due to their mobility. Each node is allocated an address based on a current subnet affiliation. Especially in large networks with random topology, it was observed that partitioning of the network may result in a substantially more balanced load than in one large multi-hop network, an attribute that can improve the network's performance significantly. The architecture is highly fault-tolerant and can handle a relatively simple location updating and tracking scheme. By virtue of its load balancing feature, the architecture typically achieves a network with relatively high throughput and low delay.

L1: Entry 16 of 16

File: USPT

Jul 29, 1997

DOCUMENT-IDENTIFIER: US 5652751 A

TITLE: Architecture for mobile radio networks with dynamically changing topology using virtual subnets

Detailed Description Text (99):

In summary, the present architecture comprises a logical topology of physical and virtual subnets, and corresponding addressing, mobility management and routing schemes. The architecture is especially applicable to mobile radio networks and accommodates dynamic topology changes due to relative movement of network nodes. The architecture partitions a mobile radio network into logically independent subnetworks. Network nodes are members of physical and virtual subnets and may change their affiliations with these subnets due to their mobility. Each node is allocated an address based on a current subnet affiliation. Especially in large networks with random topology, it was observed that partitioning of the network may result in a substantially more balanced load than in one large multi-hop network, an attribute that can improve the network's performance significantly. The architecture is highly fault-tolerant and can handle a relatively simple location updating and tracking scheme. By virtue of its load balancing feature, the architecture typically achieves a network with relatively high throughput and low delay.

WEST**End of Result Set**

Generate Collection

Print

L4: Entry 2 of 2

File: USPT

Feb 12, 2002

DOCUMENT-IDENTIFIER: US 6347078 B1

TITLE: Multiple path routing

Abstract Text (1):

A novel data structure in a router helps to compute viable next hops for forwarding a data packet from a router to its destination along multiple alternate loop-free paths, which are not necessarily of shortest distance. Each viable next hop may also be specified with a degree of optimality, which enables a route to perform QoS routing and fault-tolerant routing efficiently. The data structure can be implemented as an add-on software to existing routing protocols and may be implemented in existing networks which use shortest path protocols, even where less than all of the routers use the data structure and multiple path scheme described herein.

Current US Cross Reference Classification (3):709/105

WEST☐ Generate Collection☐ Print

L6: Entry 5 of 10

File: USPT

May 28, 2002

DOCUMENT-IDENTIFIER: US 6397260 B1

TITLE: Automatic load sharing for network routers

Brief Summary Text (8):

One way in which greater efficiency has been achieved is to implement load sharing, a technique which forces routing traffic to be distributed among a number of routers. Load sharing between routers has been known. For example, in the routing protocol OSPF routing choices depend in part on the loading at candidate next hop routers. However, OSPF is a complicated protocol that is not suitable for use by hosts or end-stations.

Brief Summary Text (9):

One way in which load sharing has been achieved in the Internet is through the configuring of so-called "default routers". A default router is a router to which a source node may send messages intended for a destination that is not directly reachable by the source. For example, the intended destination may be a node on a different subnetwork from the subnetwork on which the source node resides. The source node sends the message to the router at the default router address, and the router in turn forwards the packet toward the destination node using known techniques. Load sharing is achieved by attaching two or more routers to the network, and configuring different source nodes on a network with different default router addresses. In such a case each router handles only the portion of the overall forwarding workload that is generated by the source nodes configured with that router's address as the default router address.

Brief Summary Text (17):

When a data message is to be transmitted from a source node to a destination node on another network via a router, the source node transmits the message on the network along with the identifier returned in response to the address request message broadcast by the source node. Each router responds only to those messages that contain the router's identifier. The router responds by determining a suitable next hop node for the message and forwarding the message to the next hop node. In this manner the forwarding traffic being generated by the source nodes on the network is automatically distributed among the routers in the load-sharing set, once the source address space has been partitioned.

Detailed Description Text (2):

FIG. 1 shows a prior-art technique that provides for load sharing of a packet forwarding load. In FIG. 1, a number of hosts H1, H2, H3 and H4 are connected to an IP subnetwork labelled SUBNET 1. Also connected to the subnetwork SUBNET 1 are two routers R1 and R2. Routers R1 and R2 are also connected to a second subnetwork labelled SUBNET 2. The subnetwork SUBNET 2 may have a relatively simple structure like that of subnetwork SUBNET 1, or it may be more complex. For example, the subnetwork SUBNET 2 may be a backbone network segment connecting the subnetwork SUBNET 1 with remote subnetworks not shown in FIG. 1.

Detailed Description Text (3):

As shown in FIG. 1, router R1 is assigned an IP address which is designated IPA, and router R2 is assigned a different IP address designated IPB. Also, router R1 is assigned a MAC address designated MACA, and router R2 is assigned a MAC address designated MACB. As is known in the art, these addresses are used in communication messages sent within the subnetwork SUBNET 1 to identify the respective router R1 or

WEST

End of Result Set



Generate Collection

Print

L7: Entry 2 of 2

File: USPT

Jul 9, 2002

DOCUMENT-IDENTIFIER: US 6418139 B1

TITLE: Mechanism to guarantee quality of service to real-time traffic on IP networks

US Reference Patent Number (8):

6205146

WEST☐ Generate Collection☐ Print

L6: Entry 3 of 10

File: USPT

Jul 16, 2002

DOCUMENT-IDENTIFIER: US 6421722 B1

TITLE: Method and apparatus for providing internetworking service reliability

Drawing Description Text (5):

FIG. 4 illustrates a schematic block diagram of a plurality of networks that include sub-networks in accordance with the present invention;

Drawing Description Text (6):

FIG. 5 illustrates a schematic block diagram of an example of providing reliable internetworking services within networks having sub-networks in accordance with the present invention;

Detailed Description Text (10):

FIG. 4 illustrates a schematic block diagram of a plurality of network 82-84, operably coupled to associated NSCs 12, 14, 18 and operably coupled to MNSC 10. As shown, each of the networks 82-86 includes a plurality of sub-networks 82-1 through 82-4, 84-1 through 84-4, and 86-1 through 86-4 that are logically coupled via logical connections 89. Accordingly, the networks are not physically divided in subnetworks, but logically to provide finer granularity when re-routing services. In this embodiment, the multi-network service controller is provided with information regarding the sub-networks, such that the MNSC may establish internetworking services by defining sub-network resources to be utilized. The sub-network resources are internally controlled by the associated network service controller 12, 14, 18.

Detailed Description Text (11):

FIG. 5 illustrates an example of the MNSC providing reliable internetworking services within networks that include a plurality of sub-networks. In this illustration, the initial internetworking service is illustrated by the heavy dashed line between UNI 94 of network 82 and UNI 100 of network 84. In this illustration, a failure occurred within sub-network 84-1. Accordingly, the MNSC flags the logical connections of sub-network 84-1. As such, the logical connection of the sub-network 84-1 to NNI 96 is flagged, as is the logical connection to subnetworks 84-2, 84-3 and 84-4. Having flagged these links (i.e., intranetworking resources), the MNSC establishes a new communication path as illustrated by the solid heavy line. The internetworking resources and intranetworking resources of the newly established internetworking service are not flagged for this particular call. By comparing the illustration of the FIG. 5 with that of FIG. 2, the dividing of networks into sub-networks, provides the MNSC with greater flexibility in establishing new internetworking services. As in the example of FIG. 2, when an intranetworking resource failed within the network, the network was typically unavailable for the newly established internetworking service. In contrast, by subdividing the network, network 84 is still available for supporting the newly established internetworking service, but using different sub-network portions.

Detailed Description Text (12):

When the MNSC is establishing the new internetworking service, it follows several guidelines when working with sub-network internetworking resources. The guidelines include maintaining a list of flagged internetworking resources for each internetworking service (e.g., a call) which is segment rerouted. A link (i.e., an internetworking resource) is added to the list if it is full, is fractional, or is a logic link on the MNSC level and is adjacent to a device containing the end point of a failed segment. For example, if a DS0 is a segment endpoint, the full and all the

fractional links originating on the parent DS1 device will be flagged. If a DS0 is a frame relay segment endpoint, all the frame relay links passing through the parent DS1 device will be flagged. Note that marking of the links influences the routing only of the service (e.g., call) on whose list they are maintained and not the routing of any other service. The MNSC deletes the flag list for a service when the service becomes connected or when a certain number of segment reroute attempts have passed.

Detailed Description Text (13):

When a service has both endpoints within the same sub-network, the service cannot be segment rerouted unless the new route goes out via an NNI and into the sub-network over some internetworking links. For example, if both endpoints were contained within sub-network 82-2, and a failure occurred within the sub-network, the segment rerouting would require internetworking resources to be incorporated. For example, in FIG. 5 the internetworking resource coupled between NNI 90 and NNI 96 of network 84 would need to be utilized. Intranetworking resources of sub-network 84-1, -2, -3 and/or -4 would need to be allocated and internetworking resource between NNI 98 of network 84 and NNI 102 of network 86 would be allocated. Additionally, internetworking resources of sub-network 86-1, -2, -3, and/or -4 would need to be allocated such that the internetworking resource between NNI 92 of network 82 and NNI 104 of network 86 completes the coupling to network 82. Having completed the internetworking coupling, additional intranetworking resources of network 82 would need to be established to perform the segment rerouting.

CLAIMS:

3. The method of claim 1, wherein step (c) further comprises determining the new internetworking service based on at least one of exclusion of the at least one flagged internetworking resource, cost of the internetworking and intranetworking resources, minimal number of hops between the internetworking and intranetworking resources, load balancing of the internetworking and intranetworking resources, and bandwidth of the internetworking and intranetworking resources.

18. The multinet network service controller of claim 16, wherein the memory further comprises operational instructions that cause the processing module to determine the new internetworking service based on at least one of exclusion of the at least one flagged internetworking resource, cost of the internetworking and intranetworking resources, minimal number of hops between the internetworking and intranetworking resources, load balancing of the internetworking and intranetworking resources, and bandwidth of the internetworking and intranetworking resources.

WEST



Generate Collection

Print

L7: Entry 1 of 2

File: USPT

Nov 26, 2002

DOCUMENT-IDENTIFIER: US 6487177 B1

TITLE: Method and system for enhancing communications efficiency in data communications networks

US Reference Patent Number (7):
6205146

WEST☐ Generate Collection☐ Print

L6: Entry 1 of 10

File: USPT

Apr 15, 2003

DOCUMENT-IDENTIFIER: US 6549516 B1

TITLE: Sending instructions from a service manager to forwarding agents on a need to know basis

Detailed Description Text (11):

In addition to specifying instructions for each flow, service managers must also obtain information about each new flow from the forwarding agents. For example, when a service manager provides load balancing through a set of forwarding agents, the service manager uses fixed affinities to provide specific instructions to the forwarding agents detailing where packets for each load balanced flow are to be forwarded. In addition to providing those specific instructions, the service manager also provides general instructions to each forwarding agent that specify which new flows the service manager is interested in seeing. These general instructions are provided using wildcard affinities. Wildcard affinities, which are described in detail below, specify sets of flows that are of interest to a service manager. In one embodiment, this is done by specifying subnet masks that determine sets of source and destination IP addresses that will be forwarded to a service manager. In addition, ports or sets of ports and protocol may be specified in wildcard affinity as well. As is described further below, the use of wildcard affinities enables separate service managers to be configured to provide services for different sets of flows. Each service manager specifies the flows of interest to it and other service managers handle other flows. In this manner, service managers can be configured in parallel to share load.

Detailed Description Text (32):

The fixed affinity sent to the forwarding agent 302 may include an action that directs the forwarding agent to dispatch the SYN packet directly to host 306. The action included in the fixed affinity may also direct the forwarding agent to translate the destination address of the packet to the IP address of host 306 and the packet may be routed to host 306 via one or more hops. In addition, as described below, tag switching may also be used to send the packet to the host that is selected by the service manager using its load balancing algorithm.

Detailed Description Text (40):

FIG. 4 is a diagram illustrating a network that includes two forwarding agents and two service managers. A first client 402 and a second client 404 send packets through a network or internetwork 406 that eventually reach a subnetwork that includes a first forwarding agent 410, a second forwarding agent 412, a first service manager 420, and a second service manager 422. In the examples shown, the service managers communicate with the forwarding agents and with each other over the same physical network that is used to send packets. In other embodiments, a separate physical connection may be provided between service managers for the purpose of coordinating service managers and providing back up service managers and a separate connection may be provided between the service managers and the forwarding agents for the purpose of multicasting wildcard affinities or, in some embodiments, for sending fixed affinities and returning packets to forwarding agents.

Detailed Description Text (69):

It should be noted that in other embodiments, other methods of specifying ranges for the wildcard affinity are used. For example, in one alternative arrangement, ranges of IP addresses are specified by specifying lower bound and upper bound IP addressees. All addresses between the two bounds fall within the range of the

WEST[Generate Collection](#)[Print](#)**Search Results - Record(s) 1 through 10 of 10 returned.**☐ 1. Document ID: US 6549516 B1

L6: Entry 1 of 10

File: USPT

Apr 15, 2003

US-PAT-NO: 6549516

DOCUMENT-IDENTIFIER: US 6549516 B1

TITLE: Sending instructions from a service manager to forwarding agents on a need to know basis

DATE-ISSUED: April 15, 2003

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Albert; Mark	Wake Forest	NC		
Howes; Richard A.	Roswell	GA		
Jordan; James A.	Roswell	GA		
Kersey; Edward A.	Alpharetta	GA		
Menditto; Louis F.	Raleigh	NC		
O'Rourke; Chris	Morrisville	NC		
Tiwari; Pranav Kumar	Raleigh	NC		
Tsang; Tzu-Ming	Chapel Hill	NC		

US-CL-CURRENT: 370/236; 370/389, 370/432, 709/203, 709/226

ABSTRACT:

A system and method are described for providing instructions for forwarding packets. The method includes broadcasting a general instruction specifying a plurality of flows to a plurality of forwarding agents and receiving at a service manager a first message responsive to the general instruction indicating that a packet for a specific flow has been received by a specific forwarding agent. A specific instruction is generated at the service manager for handling the specific flow and the specific instruction for handling the specific flow is sent to the specific forwarding agent.

8 Claims, 33 Drawing figures

Exemplary Claim Number: 1

Number of Drawing Sheets: 8

L6: Entry 1 of 10

File: USPT

Apr 15, 2003

DOCUMENT-IDENTIFIER: US 6549516 B1

TITLE: Sending instructions from a service manager to forwarding agents on a need to know basis

Detailed Description Text (11):

In addition to specifying instructions for each flow, service managers must also obtain information about each new flow from the forwarding agents. For example, when

WEST Search History

DATE: Monday, July 28, 2003

Set Name Query
side by side

Hit Count Set Name
result set

DB=USPT; PLUR=YES; OP=ADJ

L6	((hop or hopping or hopped) same ((load or workload) near2 (balanc\$ or shar\$))) and (subnet\$ or (sub net\$))	10	L6
L5	6205146[pn]	1	L5
L4	l3 and hop[ti,ab]	2	L4
L3	(709/105 OR 709/102).CCLS.	817	L3
L2	(hop or hopping or hopped) same ((load or workload) near2 (balanc\$ or shar\$)) same (subnet\$ or (sub net\$))	1	L2
L1	(hop or hopping or hopped) near6 ((load or workload) near2 (balanc\$ or shar\$))	16	L1

END OF SEARCH HISTORY

WEST Search History

DATE: Monday, July 28, 2003

Set Name Query

side by side

Hit Count Set Name

result set

DB=USPT; PLUR=YES; OP=ADJ

L7	6205146[uref]	2	L7
L6	((hop or hopping or hopped) same ((load or workload) near2 (balanc\$ or shar\$))) and (subnet\$ or (sub net\$))	10	L6
L5	6205146[pn]	1	L5
L4	l3 and hop[ti,ab]	2	L4
L3	(709/105 OR 709/102).CCLS.	817	L3
L2	(hop or hopping or hopped) same ((load or workload) near2 (balanc\$ or shar\$)) same (subnet\$ or (sub net\$))	1	L2
L1	(hop or hopping or hopped) near6 ((load or workload) near2 (balanc\$ or shar\$))	16	L1

END OF SEARCH HISTORY

WEST Search History

DATE: Monday, July 28, 2003

Set Name Query

side by side

Hit Count Set Name

result set

DB=USPT; PLUR=YES; OP=ADJ

L4	l3 and hop[ti,ab]	2	L4
L3	(709/105 OR 709/102).CCLS.	817	L3
L2	(hop or hopping or hopped) same ((load or workload) near2 (balanc\$ or shar\$)) same (subnet\$ or (sub net\$))	1	L2
L1	(hop or hopping or hopped) near6 ((load or workload) near2 (balanc\$ or shar\$))	16	L1

END OF SEARCH HISTORY

WEST☐ **Generate Collection** **Print**

L1: Entry 14 of 16

File: USPT

Jun 30, 1998

DOCUMENT-IDENTIFIER: US 5774660 A

TITLE: World-wide-web server with delayed resource-binding for resource-based load balancing on a distributed resource multi-node network

Detailed Description Text (77):

When the assigned server resides on the far side of a router or across a wide-area network (WAN), the previous embodiment does not allow packets to get to their final destination. The load balancer normally transmits packets with the physical NIC address of the assigned server, but with the virtual IP address. When the next hop is not the assigned server, such as when the load balancer and the assigned server are separated by a router, the router would route the packet back to the load balancer since the packets have the load balancer's virtual IP address.

Detailed Description Text (113):

The web farm has been described as having a `local` network, but this local network could be local only in the sense that it is not the Internet backbone. Servers in the web farm may be geographically remote, where some of the servers are located in one city while other servers are located in other cities. Load balancing may be performed not just based on content, but also geographically to minimize traffic on the network backbone. The parts of the web site in a city may be connected locally through one or more LAN's, while being connected to other cities using a WAN. The IXP protocol can be used for all packets sent from the load balancer to the assigned server, even when multiple hops are not required.

L1: Entry 14 of 16

File: USPT

Jun 30, 1998

DOCUMENT-IDENTIFIER: US 5774660 A

TITLE: World-wide-web server with delayed resource-binding for resource-based load balancing on a distributed resource multi-node network

Detailed Description Text (77):

When the assigned server resides on the far side of a router or across a wide-area network (WAN), the previous embodiment does not allow packets to get to their final destination. The load balancer normally transmits packets with the physical NIC address of the assigned server, but with the virtual IP address. When the next hop is not the assigned server, such as when the load balancer and the assigned server are separated by a router, the router would route the packet back to the load balancer since the packets have the load balancer's virtual IP address.

Detailed Description Text (113):

The web farm has been described as having a `local` network, but this local network could be local only in the sense that it is not the Internet backbone. Servers in the web farm may be geographically remote, where some of the servers are located in one city while other servers are located in other cities. Load balancing may be performed not just based on content, but also geographically to minimize traffic on the network backbone. The parts of the web site in a city may be connected locally through one or more LAN's, while being connected to other cities using a WAN. The IXP protocol can be used for all packets sent from the load balancer to the assigned server, even when multiple hops are not required.

WEST☐ Generate Collection☐ Print

L1: Entry 13 of 16

File: USPT

Aug 17, 1999

DOCUMENT-IDENTIFIER: US 5940372 A

TITLE: Method and system for selecting path according to reserved and not reserved connections in a high speed packet switching network

Detailed Description Text (239):

FIG. 7 shows a general flow chart of the path selection procedure according to the present invention when the "best" path criterion is the absolute minimum path Weight (not necessarily the minimum hop). The absolute lightest path is selected independently of the hop count to obtain, for example, an efficient network load balancing.

L1: Entry 13 of 16

File: USPT

Aug 17, 1999

DOCUMENT-IDENTIFIER: US 5940372 A

TITLE: Method and system for selecting path according to reserved and not reserved connections in a high speed packet switching network

Detailed Description Text (239):

FIG. 7 shows a general flow chart of the path selection procedure according to the present invention when the "best" path criterion is the absolute minimum path Weight (not necessarily the minimum hop). The absolute lightest path is selected independently of the hop count to obtain, for example, an efficient network load balancing.

WEST

Generate Collection

Print

L1: Entry 3 of 16

File: USPT

Apr 29, 2003

DOCUMENT-IDENTIFIER: US 6556541 B1

TITLE: MAC address learning and propagation in load balancing switch protocols

Detailed Description Text (112):

The hop count field in the first element is set to 0 by the edge switch that initiates the packet and is incremented along the way by each switch the packet encounters. If the hop count gets above 0x0F, it is considered infinite and a path that cannot be taken. This prevents large topologies that may take more than 30 seconds to converge. This does not mean, however, that the topology is limited to 15 switches, but only that a path that takes more than 15 hops is not permitted within a given load balance domain. In effect, the hop count is used to limit the diameter of the network to insure convergence.

Detailed Description Text (114):

As with the other parameters, the retransmission and hop count limits may need to be adjusted as real convergence times are measured. Typically, a load balance domain topology should have a number of short hop routes and not as many long hop routes, since this adds a considerable latency and would defeat some of the benefits of the load balancing. Allowing the advanced user to specify the hop limit within a range may be advantageous, as this could be used to limit the possible number of routes and keep latency at a minimum.

WEST☐ Generate Collection☐ Print

L1: Entry 6 of 9

File: DWPI

Nov 4, 1999

DERWENT-ACC-NO: 2000-062091

DERWENT-WEEK: 200234

COPYRIGHT 2003 DERWENT INFORMATION LTD

TITLE: Network router for use in internet

Basic Abstract Text (2):

DETAILED DESCRIPTION - The output port selector maintains ordering of packets within flow by routing the packets of flow along the single route through the router fabric. The output port selector dynamically balances load across the trunks of composite trunk. The output port selector favors output ports having lesser distances to be transversed on the routing fabric from the input port, with reference to look-up table. The routing fabric used for transfer of data pack between the trunk ports is a three- dimensional torus. An INDEPENDENT CLAIM is also included for describing the packet routing method in network.

Basic Abstract Text (4):

ADVANTAGE - Appropriate setting of fabric routing table can minimize the number of hops that packets travel in routing fabric. Differential selection of nearby output trunks for each source node can be performed without concern for reordering flows, since the flows are local to single source node. By adjusting routes and hence the distribution of flows, one at a time, load incrementally approaches perfect balance across the output trunks. Simplifies routing tables by allowing large groups of destinations to be mapped to single composite output port rather than requiring that many smaller groups to be individually mapped to distinct output ports.

Equivalent Abstract Text (2):

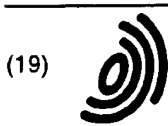
DETAILED DESCRIPTION - The output port selector maintains ordering of packets within flow by routing the packets of flow along the single route through the router fabric. The output port selector dynamically balances load across the trunks of composite trunk. The output port selector favors output ports having lesser distances to be transversed on the routing fabric from the input port, with reference to look-up table. The routing fabric used for transfer of data pack between the trunk ports is a three- dimensional torus. An INDEPENDENT CLAIM is also included for describing the packet routing method in network.

Equivalent Abstract Text (4):

ADVANTAGE - Appropriate setting of fabric routing table can minimize the number of hops that packets travel in routing fabric. Differential selection of nearby output trunks for each source node can be performed without concern for reordering flows, since the flows are local to single source node. By adjusting routes and hence the distribution of flows, one at a time, load incrementally approaches perfect balance across the output trunks. Simplifies routing tables by allowing large groups of destinations to be mapped to single composite output port rather than requiring that many smaller groups to be individually mapped to distinct output ports.

Equivalent Abstract Text (9):

DETAILED DESCRIPTION - The output port selector maintains ordering of packets within flow by routing the packets of flow along the single route through the router fabric. The output port selector dynamically balances load across the trunks of composite trunk. The output port selector favors output ports having lesser distances to be transversed on the routing fabric from the input port, with reference to look-up table. The routing fabric used for transfer of data pack



Europäisches Patentamt
European Patent Office
Office européen des brevets



(11) **EP 1 041 776 A1**

(12) **EUROPEAN PATENT APPLICATION**

(43) Date of publication:
04.10.2000 Bulletin 2000/40

(51) Int Cl.7: **H04L 12/56, H04L 29/06**

(21) Application number: **99480017.5**

(22) Date of filing: **30.03.1999**

(84) Designated Contracting States:
**AT BE CH CY DE DK ES FI FR GB GR IE IT LI LU
MC NL PT SE**
Designated Extension States:
AL LT LV MK RO SI

(71) Applicant: **INTERNATIONAL BUSINESS
MACHINES CORPORATION**
Armonk, NY 10504 (US)

(72) Inventors:
• **Lamberton, Marc**
06600 Antibes (FR)

• **Secondo, Pierre**
06140 Tourettes sur Loup (FR)
• **Levy-Abegnoli, Eric**
06200 Nice (FR)
• **Thubert, Pascal**
06140 Vence (FR)

(74) Representative: **Etorre, Yves Nicolas**
Compagnie IBM France,
Département Propriété Intellectuelle
06610 La Gaude (FR)

(54) **Multiple ARP functionality for an IP data transmission system**

(57) Data transmission system for transmitting packetized data from an IP host (10) having at least an IP layer (34) and a network layer to a plurality of workstations (12, 14) by the intermediary of an IP network (16) and wherein the IP host is connected to the IP network via a layer 2 network (18) interfacing the IP network

by a set of routers (20, 22, 24). The IP host further includes a Multiple Address Resolution Protocol (MARP) layer (36) between the IP layer and the network layer for selecting one of the set of routers in response to the next hop IP address provided by the IP layer to the multiple ARP layer when a packet of data is to be transmitted from the IP host to one of the workstations.

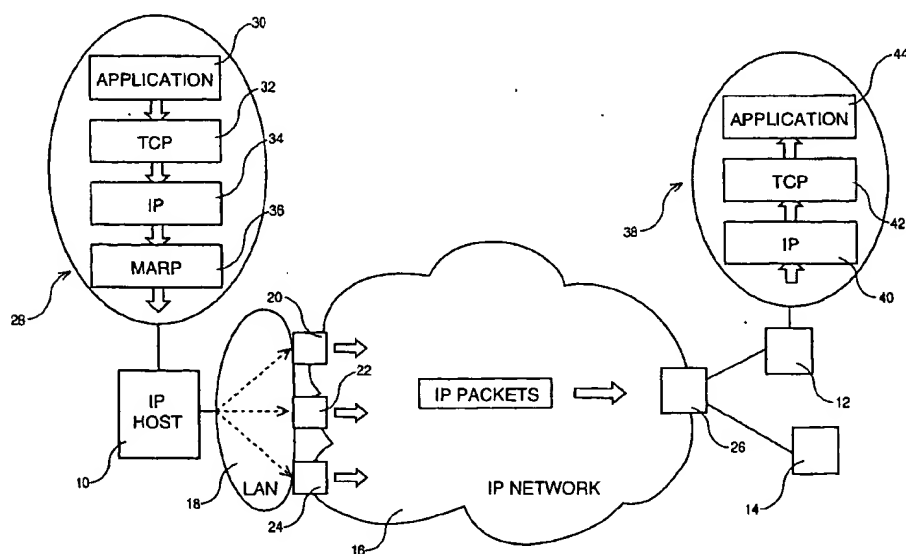


FIG. 1

EP 1 041 776 A1

WEST☐ Generate Collection☐ Print

L1: Entry 5 of 9

File: DWPI

Oct 4, 2000

DERWENT-ACC-NO: 2000-595725

DERWENT-WEEK: 200134

COPYRIGHT 2003 DERWENT INFORMATION LTD

TITLE: Method for providing multiple address resolution protocol (MARP) functionality for IP data transmission system has host with IP layer and network layer to several workstations by intermediary of IP network

Basic Abstract Text (1):

NOVELTY - The data transmission system has the IP host including a mutltiple address resolution protocol (MARP) layer (36) between the IP layer and the network layer for selecting one of a set of routers in response to the next hop IP address providing by the IP layer to the multiple ARP layer when a packet of data is to be transmitted from the IP host to one of the workstations.

Basic Abstract Text (4):

ADVANTAGE - It is the IP host which selects directly the default router thereby improving load balancing and high availability.

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau



(43) International Publication Date
20 June 2002 (20.06.2002)

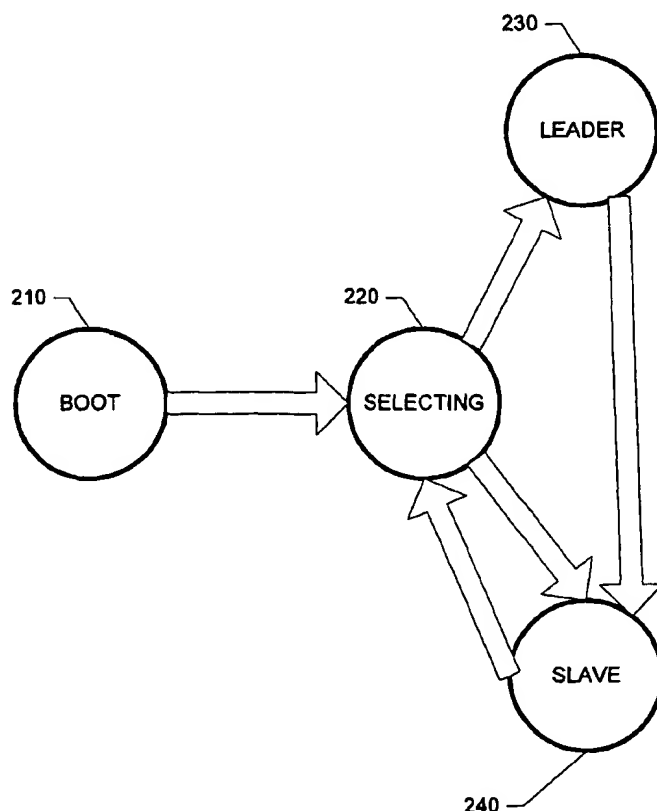
PCT

(10) International Publication Number
WO 02/48823 A2

- (51) International Patent Classification⁷: G06F [IL/IL]; 11 Hizkyhao Hamelch St., 46680 Herzliya (IL).
(21) International Application Number: PCT/IL01/01162 LEVIATAN, Chava [IL/IL]; 5 Moshe Sne St., 43728 Ra'anana (IL).
(22) International Filing Date: 13 December 2001 (13.12.2001) (81) Designated States (*national*): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, OM, PH, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZM, ZW.
(25) Filing Language: English
(26) Publication Language: English
(30) Priority Data: 60/255,075 14 December 2000 (14.12.2000) US
(71) Applicant (*for all designated States except US*): FLASH NETWORKS LTD. [IL/IL]; Bachar Avi, 16 Galgalei Hap-lada St., P.O. Box 12624, 46733 Herzliya (IL). (84) Designated States (*regional*): ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent
(72) Inventors; and
(75) Inventors/Applicants (*for US only*): SIEV, Amnon

[Continued on next page]

(54) Title: A SYSTEM AND A METHOD FOR LOAD BALANCING



(57) Abstract: The present invention is a method and system of load balancing in a group of one or more servers connected to one or more subnetworks. Two or more independent servers are bound into a group, with one of the servers elected to serve as a leader. The leader acts as a load balancer for the group while the remaining servers act as slaves. This functionality eliminates the need for one or more dedicated load balancing devices and lowers the hardware requirements necessary for performing such load balancing.

WO 02/48823 A2

WEST☐ **Generate Collection****Print**

L1: Entry 3 of 9

File: DWPI

Jun 24, 2002

DERWENT-ACC-NO: 2002-528206

DERWENT-WEEK: 200267

COPYRIGHT 2003 DERWENT INFORMATION LTD

TITLE: Transparent and symmetrical load balancing for packet-based network by using router to grab request and pretending that remote clients connect to interface

Basic Abstract Text (2):

DETAILED DESCRIPTION - IP is mapped with one of the joint IP addresses as the next Hop to a sub-network not comprising remote clients. Request reception is initiated from the networks with the IP address by the router grabbing the request, broadcasting an ARP request and forwarding the response. There is an INDEPENDENT CLAIM for a system for establishing transparent and symmetrical load balancing in a packet-based network.

Standard Title Terms (1):

TRANSPARENT SYMMETRICAL LOAD BALANCE PACKET BASED NETWORK ROUTER GRAB REQUEST REMOTE CLIENT CONNECT INTERFACE

WEST☐ Generate Collection☐ Print

L6: Entry 1 of 11

File: USPT

Feb 11, 2003

DOCUMENT-IDENTIFIER: US 6519248 B1

TITLE: Packet data network having distributed database

Detailed Description Text (82):

This description has assumed that a node knows which of its ports is a leaf-port and that a node sends out downlink packets only through this port; this avoids having downlink packets arrive at a node several times and hence eventually circulate endlessly in the network. Another way to avoid this is by modifying the format of downlink data packets, for example by including an information element similar to the time-to-live (TTL) field in conventional IP packets, i.e., a hop counter. This information element would be set to an initial value when a packet is created, and the value in that packet would be decremented by each router. Since a packet would not be able to make more hops than its initial value, the packet would not endlessly hop around the network in the event of an error. Another way to avoid this, although perhaps more costly, is to cache in each node, e.g., in each node's packet forwarding engine, the sequence numbers of downlink packets and to discard duplicates.

Detailed Description Text (89):

In fact, having two or more node trees in the same physical network has important advantages. Plural trees facilitate load sharing among nodes and selectable traffic routing, for compensating failures for example. These advantages can be realized when there is only one CAN 14 and one gateway node 16, when there is one CAN 14 and plural gateway nodes 16, and when there are plural CANs 14 and plural gateway nodes 16. One may note the special case when there are two or more gateway nodes, only one of which has a tree, the other(s) being provided for reliability purposes only. This special case can be handled in the same way as described.

WEST☐ **Generate Collection** **Print**

L6: Entry 4 of 11

File: USPT

Dec 31, 2002

DOCUMENT-IDENTIFIER: US 6502135 B1

TITLE: Agile network protocol for secure communications with assured system availability

Abstract Text (1):

A plurality of computer nodes communicate using seemingly random Internet Protocol source and destination addresses. Data packets matching criteria defined by a moving window of valid addresses are accepted for further processing, while those that do not meet the criteria are quickly rejected. Improvements to the basic design include (1) a load balancer that distributes packets across different transmission paths according to transmission path quality; (2) a DNS proxy server that transparently creates a virtual private network in response to a domain name inquiry; (3) a large-to-small link bandwidth management feature that prevents denial-of-service attacks at system chokepoints; (4) a traffic limiter that regulates incoming packets by limiting the rate at which a transmitter can be synchronized with a receiver; and (5) a signaling synchronizer that allows a large number of nodes to communicate with a central node by partitioning the communication function between two separate entities.

Brief Summary Text (16):

To transmit a data stream, a TARP originating terminal constructs a series of TARP packets from a series of IP packets generated by a network (IP) layer process. (Note that the terms "network layer," "data link layer," "application layer," etc. used in this specification correspond to the Open Systems Interconnection (OSI) network terminology.) The payloads of these packets are assembled into a block and chain-block encrypted using the session key. This assumes, of course, that all the IP packets are destined for the same TARP terminal. The block is then interleaved and the interleaved encrypted block is broken into a series of payloads, one for each TARP packet to be generated. Special TARP headers IPT are then added to each payload using the IP headers from the data stream packets. The TARP headers can be identical to normal IP headers or customized in some way. They should contain a formula or data for deinterleaving the data at the destination TARP terminal, a time-to-live (TTL) parameter to indicate the number of hops still to be executed, a data type identifier which indicates whether the payload contains, for example, TCP or UDP data, the sender's TARP address, the destination TARP address, and an indicator as to whether the packet contains real or decoy data or a formula for filtering out decoy data if decoy data is spread in some way through the TARP payload data.

Brief Summary Text (25):

Further improvements described in this continuation-in-part application include: (1) a load balancer that distributes packets across different transmission paths according to transmission path quality; (2) a DNS proxy server that transparently creates a virtual private network in response to a domain name inquiry; (3) a large-to-small link bandwidth management feature that prevents denial-of-service attacks at system chokepoints; (4) a traffic limiter that regulates incoming packets by limiting the rate at which a transmitter can be synchronized with a receiver; and (5) a signaling synchronizer that allows a large number of nodes to communicate with a central node by partitioning the communication function between two separate entities

Detailed Description Text (10):

WEST☐ Generate Collection☐ Print

L17: Entry 3 of 7

File: USPT

Apr 24, 2001

DOCUMENT-IDENTIFIER: US 6223205 B1

TITLE: Method and apparatus for assigning tasks in a distributed server system

Detailed Description Text (6):

The TCP router has been developed to provide a finer grained load balancing mechanism than was afforded by the round robin DNS approach. The TCP router modifies the destination IP address within each packet in such a way that all packets stemming from one request go to the same host, but successive requests are mapped to successive hosts in a round robin fashion. This technique allows work to be more equally distributed among the various hosts on a finer time scale than was possible with round robin DNS. Though such routers may introduce delays associated with the modification of each packet that flows through them, their benefits in terms of load balancing are such that they are commonly proposed for high performance distributed Web Servers and commercial products are beginning to appear that employ these devices.

Current US Original Classification (1):709/105Current US Cross Reference Classification (1):709/102

WEST☐ Generate Collection☐ Print

L17: Entry 4 of 7

File: USPT

Jan 30, 2001

DOCUMENT-IDENTIFIER: US 6182139 B1

TITLE: Client-side resource-based load-balancing with delayed-resource-binding using TCP state migration to WWW server farm

Abstract Text (1):

A client-side dispatcher resides on a client machine below high-level client applications and TCP/IP layers. The client-side dispatcher performs TCP state migration to relocate the client-server TCP connection to a new server by storing packets locally and later altering them before transmission. The client-side dispatcher operates in several modes. In an error-recovery mode, when a server fails, error packets from the server are intercepted by the client-side dispatcher. Stored connection packet's destination addresses are changed to an address of a relocated server. The altered packets then establish a connection with the relocated server. Source addresses of packets from the server are changed to that of the original server that crashed so that the client application is not aware of the error. In a delayed URL-based dispatch mode, the client-side dispatcher intercepts connection packets before they are sent over the network. Reply packets are faked by the client-side dispatcher to appear to be from a server and then sent to up to the client TCP/IP layers. The client's TCP then sends URL packet identifying the resource requested. The client-side dispatcher decodes the URL and picks a server and sends the packet to the server. Reply packets from the server are intercepted, and data packets altered to have the source address of the faked server. Multicast of the initial packet to multiple servers is used for empirical load-balancing by the client. The first server to respond is chosen while the others are reset. Thus the client-side dispatcher picks the fastest of several servers.

Detailed Description Text (54):

FIG. 8 highlights multicast from the client-side dispatcher to multiple servers to find the fastest-responding server to handle the request. The invention can also be used in a mode that performs an empirical load-balancing. Connection packets are sent to multiple servers. The first server to respond is likely the server with the lightest load, highest performance, or closest on the network. Thus the first server to respond is often the best choice and would be picked by a rule-based load-balancer. Indeed, such empirical load-balancing may be more efficient than traditional rule-based load balancing since a traditional load balancer's load information is slightly stale due to the latency for load queries.

Current US Cross Reference Classification (1):709/105

WEST**End of Result Set**

Generate Collection

Print

L17: Entry 7 of 7

File: USPT

Jun 30, 1998

DOCUMENT-IDENTIFIER: US 5774660 A

TITLE: World-wide-web server with delayed resource-binding for resource-based load balancing on a distributed resource multi-node network

Detailed Description Text (30):

Since load balancing is performed as soon as the first SYN packet is received, before the URL is sent, such prior-art load balancing cannot take into account the resource or file which is requested by the URL. All servers must have the same content, since the URL arrives after the server assignment has been made. The invention overcomes these limitations by delaying the assignment of the server until after the URL has been received.

Detailed Description Text (33):

FIG. 10 is a flowchart showing load balancing and state migration delayed until after the connection is made and the URL request received. The TCP connection 100 is made between the browser and the load balancer by exchanging SYN and ACK packets. Once this connection is made, the load balancer waits until the browser sends the URL which indicates which file or resource is requested. Once the URL request 102 is received, the load balancer parses the URL to determine which resource is being requested. Based on the resource requested from parsing the URL, the load balancer determines which servers are best suited to serve the request. The load balancer then performs load balancing among the servers that can serve the request, step 125.

Detailed Description Text (117):

While delayed resource binding is preferred, other embodiments are contemplated. HTTP redirection from the scheduler to the assigned server may be used after the URL has been parsed. The load balancer sends the client the address of the assigned server and instructs the client to re-issue the URL request using the assigned server's address. Thus redirection is delayed until the URL is parsed and the requested content is determined.

Current US Cross Reference Classification (3):709/105

WEST☐ Generate Collection☐ Print

L14: Entry 7 of 22

File: USPT

Oct 16, 2001

DOCUMENT-IDENTIFIER: US 6304913 B1

TITLE: Internet system and method for selecting a closest server from a plurality of alternative servers

Brief Summary Text (6):

Consequently, the service providers have developed various schemes to select a particular mirror server to service the request of a user. For example, a round robin scheme has been used where the mirror servers are assigned to address the requests of the users on a rotational basis regardless of the load on any of the mirror servers. Other more sophisticated schemes have also been used, such as load-balancing schemes which attempt to select a particular mirror server based on load distribution requirements, or timing schemes which select a particular mirror server based on time of day or day of week requirements. Unfortunately, none of the current Internet systems take into account the geographical distance or number of routers located between each mirror server and a particular user to select a mirror server (or the source server) located relatively close to the particular user. Of course, the selection of a close mirror server (or the source server) should reduce the response time required to process a request by the particular user. In addition, it would reduce the overall network load by reducing the number of routers that are traversed by the IP packets.

Detailed Description Text (5):

Generally, the Internet system 100 operates to select an alternative server (e.g., alternative server 158b) located relatively close to, or which is relatively appropriate for, a requesting host (e.g., requesting host 152a). For example, the alternative server closest to the requesting host can be selected. The selection of the closest alternative server 158b from a set of alternative servers 158b and 158e providing the same service (e.g., mirror servers) or slightly adapted variants of the same service can be based on a hop count which indicates the number of routers that a packet from the requesting host must traverse to reach a given alternative server (see FIGS. 3-11). The most appropriate alternative server will have the smallest hop count. Alternatively, the selection of the closest alternative server 158b from the set of alternative servers 158b and 158e can be done using predefined instructions and a host name of the requesting host, where the predefined instructions determine a unique Internet Protocol address of the most appropriate alternative server based on a class of the host name of the requesting host (see FIGS. 12-13). In this case, the selected alternative server does not necessarily have to be the closest. For example, the most appropriate alternative server for a requesting host with a host name ending with ".se" (the country code of Sweden) may be an alternative server using the Swedish language. A detailed description of how the selection of the closest alternative server occurs is deferred pending a discussion of the architecture of the Internet system 100.

Detailed Description Text (9):

The routers 105a-105e utilize the IP protocol to connect the respective user networks 150a-150e to the Internet 102. Each router 105a-105e contains a routing table 106 for storing hop counts derived from network topology information exchanged between the routers 105 within the Internet 102. The hop count is the total number of routers 105 and 105a-105e that an IP packet has to traverse from one of the requesting hosts 152a-152e to one of the alternative servers 158b or 158e. For example, if the requesting host 152a requested service from one of the alternative servers 158b and 158e it would take thirteen "13" hops to reach the alternative

WEST☐ Generate Collection☐ Print

L14: Entry 9 of 22

File: USPT

Jul 3, 2001

DOCUMENT-IDENTIFIER: US 6256675 B1

TITLE: System and method for allocating requests for objects and managing replicas of objects on a network

Brief Summary Text (8):

Other known commercial products offer transparent load balancing among multiple Internet sites. See CISCO Distributed Director White Paper, <http://www.cisco.com/warp/public/734/distdir/dd_wp.htm>; IBM Interactive Network Dispatcher, <<http://www.ics.raleigh.ibm.com/netdispatch/>>; Web Challenger White paper, WindDance Network Corporation, <<http://www.winddancenet.com/newwhitepaper.html>>, 1997. These products differ in the network level where the redirection of requests to physical replicas occur: CISCO's Distributed Director performs re-direction at the DNS level. A similar idea is used in E. Katz, M. Butler, and R. McGrath, A Scalable Web Server: The NCSA Prototype, Computer Networks and ISDN Systems, 27, pp. 155-164, September 1994, May 1994. The IBM Net Dispatcher and CISCO's Local Director redirect requests at the front-end router level, while Winddance's Web Challenger does so at the application level using redirection features of the HyperText Transfer Protocol (HTTP). None of these products offer dynamic replication or migration of replicas.

Brief Summary Text (13):

The works of Bestavros (A. Bestavros, Demand-based Document Dissemination to Reduce Traffic and Balance Load in Distributed Information Systems, in Proc. of the IEEE Symp. on Parallel and Distr. Processing, pp. 338-345, 1995) and Bestavros and Cunha (A. Bestavros and C. Cunha, Server-initiated Document Dissemination for the WWW, Bulletin of the Computer Society technical Committee on Data Engineering, pp. 3-11, Vol. 19, No. 3, September 1996) appear to be the predecessors of WebWave. A. Bestavros, Demand-based Document Dissemination to Reduce Traffic and Balance Load in Distributed Information Systems, in Proc. of the IEEE Symp. on Parallel and Distr. Processing, pp. 338-345, 1995 proposes to reduce network traffic within an intranet by caching organization's popular objects close to the intranet's entry point. In a very large scale system, there would be many such entry points. Such a system would address the problems of choosing entry points at which to place object replicas and allocating requests to those replicas. These questions are not considered in A. Bestavros, Demand-based Document Dissemination to Reduce Traffic and Balance Load in Distributed Information Systems, in Proc. of the IEEE Symp. on Parallel and Distr. Processing, pp. 338-345, 1995. In A. Bestavros and C. Cunha, Server-Initiated Document Dissemination for the WWW, Bulletin of the Computer Society Technical Committee on Data Engineering, pp. 3-11, Vol. 19, No. 3, September 1996, Bestavros and Cunha discuss the benefits of replicating popular objects from the host server up the request tree, but no methods for doing so are described.

Detailed Description Text (34):

Assume the availability of the following information: For any client c and any subset S of (internal) hosts, the value of BestNode (S,c); and for any client c and any host s, the (canonical) preference path between c and s. This information can be obtained efficiently in the context of actual IP routing protocols from the routes databases maintained by the routers, and hence, correspond directly to the notions of "closeness" used by the routers, currently the number of hops taken by messages en route from one node to another. As routers become more sophisticated and start using more elaborate metrics (e.g., link bandwidth, link congestion, usage fees), these metrics will be reflected in the routes databases and therefore will be picked

WEST☐

L14: Entry 10 of 22

File: USPT

Jun 26, 2001

DOCUMENT-IDENTIFIER: US 6253230 B1

TITLE: Distributed scalable device for selecting a server from a server cluster and a switched path to the selected server

Brief Summary Text (5):

The traffic on the World Wide Web (Web) is increasing exponentially, especially at popular (hot) sites. Thus it is important to provide a scaleable web server (see for example, Goldszmidt, G. and Hunt, G. "Net Dispatcher a TCP Connection Router" IBM Research Report, 1997; and Dias, D. M., Kish, W., Mukhejee, R., and Tewari, R., "A Scalable and Highly Available Web Server", Proc. 41st IEEE Computer Society Intl. Conf. (COMPCON) 1996, Technologies for the Information Superhighway, pp. 85-92, February 1996. One known method to provide load balancing in a scaleable web server is to use a so-called Network Dispatcher [see e.g., U.S. Pat No. 5,371,852, issued Dec. 6, 1994 to Attanasio et al., entitled "Method and Apparatus for Making a Cluster of Computers Appear as a Single Host," which is hereby incorporated herein by reference in its entirety; and Attanasio, Clement R. and Smith, Stephen E., "A Virtual Multi-Processor Implemented by an Encapsulated Cluster of Loosely Coupled Computers", IBM Research Report RC 18442, (1992). Here, only the address of the Network Dispatcher (ND) is given out to clients; and the Network Dispatcher distributes incoming requests among the nodes in the cluster (also called a virtual encapsulated cluster (VEC)), either in a round-robin manner, or based on the load on the nodes. In co-pending U.S. patent application Ser. No. 08/861,749, filed May 22, 1997, entitled "A Method for Local and Geographically Distributed Load Balancing Using A Generalized TCP Router", by Dias et al., which is hereby incorporated herein by reference in its entirety, an example of a generalized Network Dispatcher is disclosed, that allows routing to nodes that may be located anywhere in a general inter-network.

Detailed Description Text (22):

According to the present invention, a short-cut ATM connection is established across the network (162) for the duration of a TCP connection, so that the number of intermediate hops is minimized.

WEST☐ Generate Collection☐ Print

L14: Entry 15 of 22

File: USPT

Apr 18, 2000

DOCUMENT-IDENTIFIER: US 6052718 A
TITLE: Replica routing

Brief Summary Text (4):

Certain known approaches for automatically directing client computers to servers include, for example, round robin DNS and load balancing DNS, which direct users to one of a number of server replicas in an attempt to balance the load among many servers. In another approach called multiple hostnames, content is spread over multiple servers, each with a separate hostname. Web pages returned to users contain links that point at the replica that has been selected for the user based on load-balancing concerns and replica content considerations. In another approach called Internet load balancing, a hardware component automatically distributes user requests sent to a single IP address to one of a number of server replicas to implement load balancing. Another approach is resonant dispatch that combines load balancing with replica capability to automatically direct users to a replica that is operational, is not overloaded with requests, and contains the requested information.

Detailed Description Text (12):

At step 910, network routing table request messages are sent to all of the network routers discovered in step 905, along with any well-known or preconfigured network routers. Responses (routing tables) from the network routers are received by the client computer at step 915. At step 920 the client computer derives from the routing tables the expected performance from the client's network to all of the networks specified in the received routing tables and records this information in a network performance table. The network performance table is a list of rows, in which each row contains a network number, a net mask, and an estimate of the performance from the client to the network number (e.g., an estimate of bandwidth). A net mask (sometimes called a subnet mask) specifies which portions of an IP address contain network and subnetwork identifiers and thus should be matched to a second IP address to determine whether the two addresses are on the same network. Each network performance table entry also includes the net mask for the destination network as reported by the routing table. If no net mask is reported by a network router in a destination network, then a default net mask based on the class of the destination network's IP address (which is inferred from the initial digits of the address) is used, or another pre-specified set of rules is used. If more than one network router offers a route to a distant network, the client computer records only the best-performing route in the performance table. A single metric for replica routing "performance" is used, such as estimated bandwidth. For example, if a particular RIP network router reports the number of network hops it requires to reach a distant network, rather than the estimated bandwidth required to reach the distant network, the number of hops can be converted to estimated bandwidth by simply reducing bandwidth from an ideal fixed maximum by a fixed amount for each hop reported. Alternatively, if the address of a router on a distant network is discovered in the information received at step 915, it can be "pinged" to attempt to estimate the network performance from the client to the distant network. At step 925 if a configuration-set maximum number of iterations has not been exceeded, then at step 935 all of the network routers that were named in the routing tables received at step 915 that were not previously explored are assembled, and this set of new routers is used at step 910 to learn more about the network neighborhood. Otherwise, at step 930, internetwork performance discovery is completed, yielding a network performance table that is a list of rows, in which each row contains a network

WEST☐ Generate Collection☐ Print

L14: Entry 20 of 22

File: USPT

Dec 10, 1996

DOCUMENT-IDENTIFIER: US 5583991 A

TITLE: Method for providing for automatic topology discovery in an ATM network or the like

Brief Summary Text (28):

It is also noted that in certain prior art topology discovery methods, topology information is transmitted in the form of messages to, for example, a central management unit which is capable of receiving, storing and displaying the topology information. In such systems, the topology message may be altered, to include additional information on the transmission path, as it is transmitted from device to device until it finally reaches the central management unit. In such systems, the topology message length is dependent on the number of hops between the originally transmitting device and the central manager or on the number of nodes in the network or on the number of neighbor nodes. Of course, in systems, such as ATM systems, offering a limited message size per transmitted cell, this offers disadvantages.

Detailed Description Text (138):

In order to provide for increased reliability and service redundancy, among other advantages, the described system provides for the concept of hunt groups. Membership in a hunt group may be based on one or more of a number of different criteria--namely, membership can be based on the client's address, service type, and/or resource type. The concept may be generally thought of as a process wherein a client registers as a member of a particular hunt group. Then, when the CMS attempts to set up a connection to that client, if the client is busy, unreachable (e.g., been removed from the network), or otherwise can't service the particular request, the call is redirected, transparent to the requesting client, to an alternative member of the hunt group. It is noted that this concept could be readily extended, for example to provide load-balancing among resources such as amongst a group of servers in a network. A requesting client may request, from the CMS, a connection with a particular device which has registered as a member of a particular hunt group. In this scenario, if the particular device is unavailable, the CMS may select an alternative device from the same hunt group and set up the connection with that device. Alternatively, the requesting device may request, again from the CMS, the services of any device which is a member of a designated hunt group. In this case, the CMS selects a device from the hunt group with which to establish the connection. The selection may be made based on criteria to provide an optimal connection, e.g., to achieve load balancing.

Detailed Description Text (163):

The booting switch, after receiving the PAR messages on various ports (a PAR message will have been received on each port which is coupled to a booted neighbor switch), chooses a port to use for the rest of the boot process, block 2003. As will be described in connection with the description of the various message formats, given below, the PAR message includes a field titled CMS cost factor (field 2126) which may, for example, include the number of hops between the neighbor switch attached to the port on which the PAR was received and the CMS. This field is calculated and provided by the CMS based on its tables. It is noted that, although this information is provided to the booting switch, and the booting switch may use it as a factor or the factor in determining which port to use of booting operations, selection of the particular port for use is switch dependent. Any number of algorithms may be employed.

L14: Entry 20 of 22

File: USPT

Dec 10, 1996

WEST**End of Result Set**

Generate Collection

Print

L14: Entry 22 of 22

File: USPT

Feb 14, 1995

DOCUMENT-IDENTIFIER: US 5390170 A

TITLE: Method and apparatus providing for bootstrapping of switches in an ATM network or the like

Brief Summary Text (28):

It is also noted that in certain prior an topology discovery methods, topology information is transmitted in the form of messages to, for example, a central management unit which is capable of receiving, storing and displaying the topology information. In such systems, the topology message may be altered, to include additional information on the transmission path, as it is transmitted from device to device until it finally reaches the central management unit. In such systems, the topology message length is dependent on the number of hops between the originally transmitting device and the central manager or on the number of nodes in the network or on the number of neighbor nodes. Of course, in systems, such as ATM systems, offering a limited message size per transmitted cell, this offers disadvantages.

Detailed Description Text (135):

In order to provide for increased reliability and service redundancy, among other advantages, the described system provides for the concept of hunt groups. Membership in a hunt group may be based on one or more of a number of different criteria--namely, membership can be based on the client's address, service type, and/or resource type. The concept may be generally thought of as a process wherein a client registers as a member of a particular hunt group. Then, when the CMS attempts to set up a connection to that client, if the client is busy, unreachable (e.g., been removed from the network), or otherwise can't service the particular request, the call is redirected, transparent to the requesting client, to an alternative member of the hunt group. It is noted that this concept could be readily extended, for example to provide load-balancing among resources such as amongst a group of servers in a network. A requesting client may request, from the CMS, a connection with a particular device which has registered as a member of a particular hunt group. In this scenario, if the particular device is unavailable, the CMS may select an alternative device from the same hunt group and set up the connection with that device. Alternatively, the requesting device may request, again from the CMS, the services of any device which is a member of a designated hunt group. In this case, the CMS selects a device from the hunt group with which to establish the connection. The selection may be made based on criteria to provide an optimal connection, e.g., to achieve load balancing.

Detailed Description Text (160):

The booting switch, after receiving the PAR messages on various ports (a PAR message will have been received on each port which is coupled to a booted neighbor switch), chooses a port to use for the rest of the boot process, block 2003. As will be described in connection with the description of the various message formats, given below, the PAR message includes a field titled CMS cost factor (field 2126) which may, for example, include the number of hops between the neighbor switch attached to the port on which the PAR was received and the CMS. This field is calculated and provided by the CMS based on its tables. It is noted that, although this information is provided to the booting switch, and the booting switch may use it as a factor or the factor in determining which port to use of booting operations, selection of the particular port for use is switch dependent. Any number of algorithms may be employed.

WEST**End of Result Set**

Generate Collection

Print

L12: Entry 8 of 8

File: USPT

Jun 20, 2000

DOCUMENT-IDENTIFIER: US 6078953 A

TITLE: System and method for monitoring quality of service over network

Detailed Description Text (3):

Network QoS occurs by managing the resources that serve network application traffic, for example. This typically includes the following resources: link bandwidth, application server bandwidth (CPU), and buffer space on generally all nodes (end-points, routers and gateways). Typically, data through-put is limited by the speed of Internet access links and by the server CPU capacity, and response time is determined by the number of hops in a route, physical length of the route, and extent of congestion in the route. There are various other factors that may affect QoS, such as the behavior of TCP/IP, severe congestion anywhere in the route, prioritization of traffic along the route, etc. To a network administrator, embodiments of the present invention provide discrimination of different traffic types and provide methods for enforcement of traffic flow by management to the above resources.

Detailed Description Text (30):

The present invention takes into account, in one or more embodiments, the factors which are described specifically above. Although the above has been generally described in terms of a specific type of information, other types of information on a network can also be used with the present invention. Additionally, the present invention has been described in general to a specific system. For instance, the present bandwidth management tool can be applied at a network's Internet access link. Alternatively, the present tool can be applied to a private WAN link to a remote corporate site or an access to a server farm (e.g., a group of servers located in a special part of the network close to an access link, e.g., in a web hosting environment). Alternatively, the present invention can be applied to key servers (e.g., database/web server) within an organization servicing internal and/or external users. Furthermore, the present bandwidth management tool can be applied to any combination of the above or the like.

WEST☐ Generate Collection☐ Print

L10: Entry 1 of 20

File: USPT

Dec 31, 2002

DOCUMENT-IDENTIFIER: US 6502125 B1

TITLE: System and method for optimized storage and retrieval of data on a distributed computer network

Detailed Description Text (7):

Several delivery, or "mirror" sites are shown connected to the Internet 10 in FIG. 1. A first delivery site 26 might be located a small number of "hops" from the first user terminal 12. A second delivery site 28 might be located further away from the first user terminal 12, but close to the third user terminal 20. A third delivery site 30 might be as close to the third user terminal 20 as the second delivery site 28 is. As previously noted, a user and a provider or delivery site that are "geographically" near each other might not be "electronically" near each other on the Internet. By decreasing the "electronic" distance between the user and the provider or delivery site, the number of network connections and routers over which data must travel can be decreased.

Detailed Description Text (30):

Preferably, testing should not contribute more than approximately 5% of total server load. One way to reach this goal is to lightly test a large number of servers, yielding a group of delivery sites having adequate performance. This group of delivery sites can then be used in rotation to retrieve data. Information on multimedia clip actual download times for each of the delivery sites in the group is accumulated as discussed below, and further information on delivery site performance can then be furnished to the MSP 32 transparently, without the need for further outright testing.

Detailed Description Text (45):

The player program first analyzes the EMBED tag to determine if there is an "SM" (Smart Mirror) parameter (step 60); the presence of such a parameter indicates that the embedded clip is enabled for Smart Mirroring. Data associated with the "SM" parameter specifies the particular content provider from which the desired clip originated, as well as the group of mirror servers that particular content provider uses.

Other Reference Publication (6):

Bestavros, Demand-based Document Dissemination to Reduce Traffic and Balance Load in Distributed Information Systems, Proceedings of SPDP '95: The 7th Symposium on Parallel and Distributed Processing, San Antonio, Texas, Oct. 1995.

WEST☐ Generate Collection☐ Print

L10: Entry 2 of 20

File: USPT

Nov 19, 2002

DOCUMENT-IDENTIFIER: US 6484143 B1

TITLE: User device and system for traffic management and content distribution over a world wide area network

Brief Summary Text (9):

In a specific embodiment, the invention provides a service based system for traffic management and content distribution for a plurality of users over a world wide network of computers. The system includes a global traffic management device coupled to a world wide area network. The global traffic management device being provided to load balance across multiple origin sites. The system also has a content delivery network coupled to the global traffic management device. The content delivery network provides support content distribution and delivery of streaming media. The system also has a computing device including a computer memory coupled to the global traffic management device. The system also has an accounting module coupled to the computing device. The accounting module tracks a usage of the global traffic management device and the content delivery network for a customer of the global... traffic management device and the content delivery network to determine a service fee for the usage based upon a period time frequency.

Detailed Description Text (5):

Customers can leverage the size, scope, and location of the UDN to store content such as HTML, images, video, sound and software for fast and highly available access by clients. The network can also incorporate customer origin sites 107, 109 that will then benefit from shared load balancing and traffic management. Customers with generated content, such as search engines, auctions and shopping carts, can use the latter feature to add their own content servers to the network. In some embodiments, the system typically requires no software or hardware to be installed or run at a customer site. A web interface is available for display of the network's current status as well as historical statistics on a per customer basis.

Detailed Description Text (10):

Multiple DNS servers are deployed to provided high availability. The DNS servers are spread throughout the network to avoid single points of failure. The DNS server was designed from the beginning with the ability to proxy requests. This proxy ability combined with algorithms to divide client latency and persistence information across a group of DNS servers greatly reduces the problems associated with WAN replication and synchronization. In the event a request arrives at a DNS server that is not authoritative for this client, the DNS can proxy the request to any number of servers to find an authoritative answer.

Detailed Description Text (15):

The HTTP service is an example of the service test approach. Rather than try to test the individual characteristics of a server that may have an impact on performance, the service itself is evaluated as a user would experience it, in order to determine its response time and validity. LOADP, a process running on each server, is implemented as a statistical monitor and is used as a generic service for testing purposes. LOADP provides direct measurement of many system parameters including CPU load, memory usage, swap and disk status, and is used in load balancing decisions.

Detailed Description Text (42):

In a specific embodiment, the present network includes one or more services. Here, the network may offer services, including: 1. Global Traffic Management--Provides

WEST☐ Generate Collection☐ Print

L10: Entry 4 of 20

File: USPT

Dec 11, 2001

DOCUMENT-IDENTIFIER: US 6330671 B1

TITLE: Method and system for secure distribution of cryptographic keys on multicast networks

Abstract Text (1):

A method and apparatus for secure and scalable key management in a multicast network environment is provided. In a first portion, one or more seed nodes on the network receive a multicast transmission request for a cryptographic key from a requesting node. The seed node compares the identity of the requesting node with an authenticated predetermined list of nodes having permission to receive the cryptographic key. If the comparison indicates the requesting node is not a member of the authenticated predetermined list, the seed node denies the multicast request. However, if the comparison indicates that the requesting node is a member of the predetermined list of nodes, the cryptographic key is transmitted using a secure unicast key distribution technique such as SKIP. A second portion concerns the requesting node which generates a multicast request to obtain the cryptographic key from one or more seed nodes and one or more keyed nodes on the internetwork. The multicast request for the cryptographic key is initially transmitted a minimum hop count over the internetwork to locate the closest seed node. The requesting node delays a brief time period waiting for at least one response from at least one seed node or keyed node on the internetwork. If the at least one response is not received within this time period, the minimum hop count is increased by a hop count increment and the requesting node repeats the above steps. Eventually, the requesting node increases the hop count and receives the cryptographic key over a secure unicast key management technique such as SKIP. As a final step, the requesting node is converted into a keyed node. The keyed node acts as a seed node and provides the cryptographic key to other requesting nodes on the internetwork.

Brief Summary Text (13):

Distribution of the traffic key in multicast SKIP scales well to large multicast groups because each member of the multicast group receiving a packet has a copy of the traffic key. However, the distribution of the GIK does not scale as easily. Unlike the inline traffic key, SKIP uses public-key key distribution to distribute the GIK with a single server known as the Group Controller or GC. When membership in the multicast group gets too large and geographically spread out, the GC can have difficulty distributing the GIK. Thus, while traffic key distribution using an inline traffic key is highly scalable, the distribution of the GIK from a single GC remains only moderately scalable. Details on multicast SKIP are included in the paper entitled "Design and Implementation of SKIP", authored by Ashar Aziz and Martin Patterson, Jun. 28, 1995.

Brief Summary Text (23):

In accordance with the second embodiment, the requesting node generates a multicast request to obtain the multicast cryptographic key from one or more seed nodes and one or more keyed nodes on the internetwork. The multicast request for the multicast cryptographic key is initially transmitted a minimum hop count over the internetwork to locate the closest seed node. The hop count provides an upper limit on the number of networks the request will traverse to locate a seed node. The requesting node delays a brief time period waiting for at least one response from at least one seed node or keyed node on the internetwork. If the at least one response is not received within this time period, the minimum hop count is increased by a hop count increment and the requesting node repeats the requesting process above. By increasing the hop

WEST☐ Generate Collection☐ Print

L10: Entry 6 of 20

File: USPT

Dec 4, 2001

DOCUMENT-IDENTIFIER: US 6327622 B1

TITLE: Load balancing in a network environmentAbstract Text (1):

A method is provided for load balancing requests for an application among a plurality of instances of the application operating on a plurality of servers. A policy is selected for choosing a preferred server from the plurality of servers according to a specified status or operational characteristic of the application instances, such as the least-loaded instance or the instance with the fastest response time. The policy is encapsulated within multiple levels of objects or modules that are distributed among the servers offering the application and a central server that receives requests for the application. A first type of object, a status object, gathers or retrieves application-specific information concerning the specified status or operational characteristic of an instance of the application. Status objects interact with instances of the load-balanced application and are configured to store their collected information for retrieval by individual server monitor objects. An individual server monitor object illustratively operates for each server operating an instance of the application and retrieves the application-specific information from one or more status objects. A central replicated monitor object gathers the information from the individual server monitor objects. The information is then analyzed to select the server having the optimal status or operational characteristic. An update object updates the central server, such as a domain name server, to indicate the preferred server. Requests for the application are then directed to the preferred server until a different preferred server is identified.

Parent Case Text (1):

U.S. Pat. No. 6,092,178, entitled "Systems for Responding to a Resource Request," and U.S. patent application Ser. No. 09/146,848, entitled "Load Balancing for Replicated Services," both of which were filed on Sep. 3, 1998, are related to the present application.

Brief Summary Text (2):

This invention relates to the field of computer systems. More particularly, a system and methods are provided for load balancing among application programs or replicated services.

Brief Summary Text (4):

A service offered simultaneously on multiple servers is often termed "replicated" in recognition of the fact that each instance of the service operates in substantially the same manner and provides substantially the same functionality as the others. The multiple servers may, however, be situated in various locations and serve different clients. Application programs may also operate simultaneously on multiple servers, with each instance of an application operating independently of, or in concert with, the others. In order to make effective use of an application or replicated service offered by multiple servers (e.g., to satisfy clients' requests), there must be a method of distributing clients' requests among the servers and/or among the instances of the application or service. This process is often known as load balancing. Methods of load balancing among instances of a replicated service have been developed, but are unsatisfactory for various reasons.

Brief Summary Text (5):

WEST☐ Generate Collection☐ Print

L10: Entry 10 of 20

File: USPT

Dec 26, 2000

DOCUMENT-IDENTIFIER: US 6167438 A

TITLE: Method and system for distributed caching, prefetching and replication

Detailed Description Text (22):

To accomplish this load management, or load balancing, the resource manager 24 maintains information about the identity and the load of its neighboring cache servers 30. The details of how neighboring cache server information is maintained is discussed below in Section 3.

Detailed Description Text (24):

Other responsibilities of the resource manager 24 include neighborhood discovery, propagating load information to the neighboring servers 30, and discovering and recovering from potential barriers to load balancing. These mechanisms are discussed in more detail below.

Detailed Description Text (53):

However, any resulting changes in the configuration of adjacent cache servers must also be detected by communication with neighboring cache servers in order to achieve resource load balancing and other advantages possible with the invention. In particular, each cache server 16 participating in the above-described scheme has to determine which other servers are in its neighborhood. In addition, on each routing tree T, a cache server 16 has to distinguish between upstream servers (located at parent nodes) and down stream servers (located at child nodes). A particular node, i, in the tree T is the parent of a node j, if i is the first cache server 16 on the route from j to the home server 20, in which case node j is also referred to as the child of node i.

Detailed Description Text (55):

These neighborhood discovery packets are then intercepted by a given snooper at another node having a cache server 16 in the tree. It is then responsibility of the intercepting cache server 16 to send a reply to the resource manager 24 at the cache server 16 that issued the neighborhood discover packet, announcing that it is a parent (e.g., that it is closer to the home server 20 than the issuing cache server) and the identity of the tree T that it is on. The destination port for neighborhood discover packets may be assigned an unlikely port number, to ensure that the destination home server 20 does not attempt to process un-intercepted neighborhood packets. A hop count field can also be used to limit neighborhood discover packets from excessive forwarding.

Detailed Description Text (64):4. Load BalancingDetailed Description Text (66):

The above scheme of document caching and neighborhood discovery lends itself to a number of different types of such cache load distribution and/or load balancing objectives for both the cache servers 16 as well as the communication paths which interconnect them. In the preferred embodiment, this load distribution scheme attempts to avoid introducing an overhead that grows quickly with the size of the caching system, by using a diffusion based caching algorithm that relies strictly on local information.

Detailed Description Text (99):

WEST☐ Generate Collection☐ Print

L10: Entry 11 of 20

File: USPT

Aug 29, 2000

DOCUMENT-IDENTIFIER: US 6112239 A

TITLE: System and method for server-side optimization of data delivery on a distributed computer network

Brief Summary Text (42):

In general, an improved delivery site for a particular user can be predicted in advance by analyzing aggregate network performance data collected from network tests previously performed by a group of users. Thus, delivery site selection can occur on-the-fly each time the user requests a file managed by the mirror service provider's delivery system. From the perspective of the user, the selection of the delivery site happens automatically and transparently such that there appears to be no delay between selecting a file from a web page and having the file delivered to the user's terminal. The look-up list maintained by the service provider is constantly updated to reflect changes in network performance, making it possible for the service provider to effectively load-balance network traffic.

Detailed Description Text (7):

Several delivery, or "mirror" sites are shown connected to the Internet 10 in FIG. 1. A first delivery site 26 might be located a small number of "hops" from the first user terminal 12. A second delivery site 28 might be located further away from the first user terminal 12, but close to the third user terminal 20. A third delivery site 30 might be as close to the third user terminal 20 as the second delivery site 28 is. As previously noted, a user and a provider or delivery site that are "geographically" near each other might not be "electronically" near each other on the Internet. By decreasing the "electronic" distance between the user and the provider or delivery site, the number of network connections and routers over which data must travel can be decreased.

Detailed Description Text (55):

Preferably, testing should not contribute more than approximately 5% of total server load. One way to reach this goal is to lightly test a large number of servers, yielding a group of delivery sites having adequate performance. This group of delivery sites can then be used in rotation to retrieve data. Information on multimedia clip actual download times for each of the delivery sites in the group is accumulated as discussed below, and further information on delivery site performance can then be furnished to the MSP 32 transparently, without the need for further outright testing.

Detailed Description Text (70):

The player program first analyzes the EMBED tag to determine if there is an "SM" (Smart Mirror) parameter (step 60); the presence of such a parameter indicates that the embedded clip is enabled for Smart Mirroring. Data associated with the "SM" parameter specifies the particular content provider from which the desired clip originated, as well as the group of mirror servers that particular content provider uses.

Detailed Description Text (99):

By modifying the look-up table, the redirection server can also perform load balancing and management with respect to file requests. If the MSP 32 (or an individual controlling the MSP 32) determines in advance that sections of the network will be down for a period of time, or if certain delivery sites must be shut down for a period of time, the look-up table can be modified so that the redirection

WEST☐ Generate Collection☐ Print

L10: Entry 12 of 20

File: USPT

Aug 22, 2000

DOCUMENT-IDENTIFIER: US 6108727 A

TITLE: System having wireless interface device for storing compressed predetermined program files received from a remote host and communicating with the remote host via wireless link

Parent Case Text (4):

This case is also related to the following cases, all filed on even date: MULTIPLE WIRELESS INTERFACES TO A SINGLE SERVER, by S. C. Gladwin, A. Soucy and J. Wilson, Ser. No. 08/783,708; WIRELESS ENUMERATION OF AVAILABLE SERVERS, by S. C. Gladwin, D. Bi, A. Gopalan, and J. Wilson, Ser. No. 08/784,275; DYNAMIC SERVER ALLOCATION FOR LOAD BALANCING WIRELESS INTERFACE PROCESSING, by D. Bi, Ser. No. 08/784,211; DATA COMPRESSION LOADER, by D. Boals and J. Wilson, Ser. No. 08/783,080; MULTI-USER RADIO FLASH ROM UPDATE, by D. Bi and J. Wilson, Ser. No. 08/784,141; AUDIO COMPRESSION IN A WIRELESS INTERFACE DEVICE, by S. C. Gladwin, D. Bi and D. Voegeli, Ser. No. 08/784,243; MULTI-USER ON-SCREEN KEYBOARD, by D. Bi, Ser. No. 08/784,034; LOCAL HANDWRITING RECOGNITION IN A WIRELESS INTERFACE TABLET, by S. C. Gladwin, D. Bi, D. Boals and J. Wilson, Ser. No. 08/784,034; INK TRAILS ON A WIRELESS REMOTE INTERFACE TABLET, by S. C. Gladwin, D. Bi, D. Boals, J. George, S. Merkle and J. Wilson, Ser. No. 08/784,688, and MODE SWITCHING FOR PEN-BASED COMPUTER SYSTEMS, by D. Bi, Ser. No. 08/784,212.

Detailed Description Text (275):

Referring to FIG. 69, after the server name and node address information is received by the wireless interface device 100, an IPX packet is directed to the server 1708, 1710 to request the domain name, software version, as indicated in step 1740. (Steps 1740 and 1746 may also include information whether a particular application is supported, which is part of a load balancing function described below.) The IPX packet is received by the server 1708, 1710, which, in turn, requests its domain name, as illustrated in steps 1742 and 1744. In a server running the Windows NT operating system, all domain names must be authenticated to a primary domain controller. The server then sets up packets identifying its server domain name and software version, in step 1746. This information is returned to the wireless interface device 100 and then put into a server list buffer in step 1748 and displayed in the dialog box 1732 (FIG. 70). Control is then transferred to the client manager for the wireless interface device 100 in step 1750 in the wireless interface device 100. The wireless interface device 100 may be then connected to the selected server by depressing the connect button 1738 in the set-up dialog box illustrated in FIG. 70.

Detailed Description Text (276):

The SAP query packet is initiated by way of the wireless interface device 100 by way of the set-up dialog box illustrated in FIG. 70. As discussed above, the set-up dialog box can be accessed by depressing the hot icon 1410 (FIG. 37) in the hot icon area 1202 (FIG. 36) of the wireless interface device 100. As illustrated in FIG. 70, the set-up dialog box includes a server button 1728, as well as dialog boxes 1730 and 1732 which identify the server domain names, as well as server name for those servers which broadcast a SAP advertising packet. The set-up dialog box also includes a disconnect button 1734 and update list dialog button 1736, as well as a connect button 1738. In order for the wireless interface device 100 to issue a SAP query packet, as discussed above, the update list button 1736 on the set-up dialog box is depressed. As mentioned above, the servers 1708, 1710 then return their server names and node addresses 1708, 1710, on the network. This information is

WEST☐ Generate Collection☐ Print

L10: Entry 13 of 20

File: USPT

Jul 18, 2000

DOCUMENT-IDENTIFIER: US 6092178 A

TITLE: System for responding to a resource request

Abstract Text (1):

A trigger is provided in association with a network naming service, such as DNS (Domain Name Service), that handles client requests for an application. The trigger comprises a set of executable instructions referenced by a resource record associated with an identifier of the application. In response to a client request concerning the application, the resource record is retrieved and the instructions are executed. In one implementation of a trigger, a DNS server provides load balancing among a plurality of servers within a network name space (e.g., domain or sub-domain) offering an application program (or replicated service) that is known by a virtual server name. A policy is selected for choosing a preferred server from the plurality of servers according to a specified status or operational characteristic of the application instances, such as the least-loaded instance of the application or the instance with the fastest response time. The policy is encapsulated within multiple levels of objects or modules distributed among the plurality of servers and the DNS server. The objects collect and assemble the servers' status and operational characteristics. The information collected by the objects is analyzed to select the server that best satisfies the selected policy. A client request for the application is received by the DNS server, which retrieves a resource record corresponding to the virtual server name. Within the record is the name of a trigger. The trigger is executed to select, or retrieve an identity of, a server to which the client request is to be directed.

Parent Case Text (1):

The following co-pending U.S. patent applications, filed on Sep. 3, 1998, are related to the present application and are hereby incorporated by reference: application Ser. No. 09/146,772, entitled "Load Balancing in a Network Environment," and application Ser. No. 09/146,848, entitled "Load Balancing for Replicated Services."

Brief Summary Text (4):

Multiple servers may be configured to offer an application or replicated service (e.g., a service offered simultaneously on each of multiple servers), in which case the client may be directed to any of the multiple servers in order to satisfy the client's request. In addition, the multiple servers may be situated in various locations and/or serve different clients. Therefore, in order to make effective use of the application or replicated service, a method is needed to distribute clients' requests among the servers and/or among the instances of the application or service. This process is often known as load balancing.

Brief Summary Text (5):

DNS servers can be configured to provide load balancing in addition to their traditional roles (e.g., resolving requests for information concerning a network entity). Methods of using a DNS server for load balancing among instances of a replicated service have been developed, but are unsatisfactory for various reasons.

Brief Summary Text (6):

In one method of load balancing a replicated service, a DNS server directs or assigns requests to the servers offering the service on a round-robin

WEST☐ Generate Collection☐ Print

L10: Entry 14 of 20

File: USPT

Mar 17, 1998

DOCUMENT-IDENTIFIER: US 5729685 A

TITLE: Apparatus for determining the topology of an ATM network or the like Via communication of topology information between a central manager and switches in the network over a virtual service path

Brief Summary Text (28):

It is also noted that in certain prior art topology discovery methods, topology information is transmitted in the form of messages to, for example, a central management unit which is capable of receiving, storing and displaying the topology information. In such systems, the topology message may be altered, to include additional information on the transmission path, as it is transmitted from device to device until it finally reaches the central management unit. In such systems, the topology message length is dependent on the number of hops between the originally transmitting device and the central manager or on the number of nodes in the network or on the number of neighbor nodes. Of course, in systems, such as ATM systems, offering a limited message size per transmitted cell, this offers disadvantages.

Detailed Description Text (137):

In order to provide for increased reliability and service redundancy, among other advantages, the described system provides for the concept of hunt groups. Membership in a hunt group may be based on one or more of a number of different criteria--namely, membership can be based on the client's address, service type, and/or resource type. The concept may be generally thought of as a process wherein a client registers as a member of a particular hunt group. Then, when the CMS attempts to set up a connection to that client, if the client is busy, unreachable (e.g., been removed from the network), or otherwise can't service the particular request, the call is redirected, transparent to the requesting client, to an alternative member of the hunt group. It is noted that this concept could be readily extended, for example to provide load-balancing among resources such as amongst a group of servers in a network. A requesting client may request, from the CMS, a connection with a particular device which has registered as a member of a particular hunt group. In this scenario, if the particular device is unavailable, the CMS may select an alternative device from the same hunt group and set up the connection with that device. Alternatively, the requesting device may request, again from the CMS, the services of any device which is a member of a designated hunt group. In this case, the CMS selects a device from the hunt group with which to establish the connection. The selection may be made based on criteria to provide an optimal connection, e.g., to achieve load balancing.

Detailed Description Text (143):

FIG. 17 is useful for providing an overview of an exemplary network employing hunt groups. FIG. 17 illustrates a plurality of clients 1701-1708 coupled in communication with an ATM cloud 1721. In the illustrated network, clients C3, C6, and C8 (1703, 1706 and 1708, respectively) are workstations. These workstations may require access to various network resources, such as file servers, multicast servers and printers. Clients C1 and C7 (1701 and 1707, respectively) are printers and clients C2, C4 and C5 (1702, 1704 and 1705, respectively) are file servers. In this network, for example, a printer hunt group may exist and the various printers may join the printer hunt group and a file server hunt group may also exist. Assume that file servers C2 and C5 (1702 and 1705) are used to store identical copies of applications software and various databases, while server C4 1704 stores other information unique to it. In this case, file servers C2 and C5 may both choose to

WEST**End of Result Set**

Generate Collection

Print

L10: Entry 20 of 20

File: USPT

Feb 14, 1995

DOCUMENT-IDENTIFIER: US 5390170 A

TITLE: Method and apparatus providing for bootstrapping of switches in an ATM network or the like

Brief Summary Text (28):

It is also noted that in certain prior art topology discovery methods, topology information is transmitted in the form of messages to, for example, a central management unit which is capable of receiving, storing and displaying the topology information. In such systems, the topology message may be altered, to include additional information on the transmission path, as it is transmitted from device to device until it finally reaches the central management unit. In such systems, the topology message length is dependent on the number of hops between the originally transmitting device and the central manager or on the number of nodes in the network or on the number of neighbor nodes. Of course, in systems, such as ATM systems, offering a limited message size per transmitted cell, this offers disadvantages.

Detailed Description Text (135):

In order to provide for increased reliability and service redundancy, among other advantages, the described system provides for the concept of hunt groups. Membership in a hunt group may be based on one or more of a number of different criteria--namely, membership can be based on the client's address, service type, and/or resource type. The concept may be generally thought of as a process wherein a client registers as a member of a particular hunt group. Then, when the CMS attempts to set up a connection to that client, if the client is busy, unreachable (e.g., been removed from the network), or otherwise can't service the particular request, the call is redirected, transparent to the requesting client, to an alternative member of the hunt group. It is noted that this concept could be readily extended, for example to provide load-balancing among resources such as amongst a group of servers in a network. A requesting client may request, from the CMS, a connection with a particular device which has registered as a member of a particular hunt group. In this scenario, if the particular device is unavailable, the CMS may select an alternative device from the same hunt group and set up the connection with that device. Alternatively, the requesting device may request, again from the CMS, the services of any device which is a member of a designated hunt group. In this case, the CMS selects a device from the hunt group with which to establish the connection. The selection may be made based on criteria to provide an optimal connection, e.g., to achieve load balancing.

Detailed Description Text (141):

FIG. 17 is useful for providing an overview of an exemplary network employing hunt groups. FIG. 17 illustrates a plurality of clients 1701-1708 coupled in communication with an ATM cloud 1721. In the illustrated network, clients C3, C6, and C8 (1703, 1706 and 1708, respectively) are workstations. These workstations may require access to various network resources, such as file servers, multicast servers and printers. Clients C1 and C7 (1701 and 1707, respectively) are printers and clients C2, C4 and C5 (1702, 1704 and 1705, respectively) are file servers. In this network, for example, a printer hunt group may exist and the various printers may join the printer hunt group and a file server hunt group may also exist. Assume that file servers C2 and C5 (1702 and 1705) are used to store identical copies of applications software and various databases, while server C4 1704 stores other information unique to it. In this case, file servers C2 and C5 may both choose to

WEST Search History

DATE: Thursday, March 27, 2003

Set Name Query

side by side

Hit Count Set Name

result set

DB=USPT; PLUR=YES; OP=ADJ

L10	l7 and l9	20	L10
L9	load\$ near2 balanc\$	9995	L9
L8	l3 same l6	5	L8
L7	l3 and l6	45	L7
L6	server near2 (group\$ or farm\$)	1147	L6
L5	l4 and l3	8	L5
L4	(709/105 OR 709/102).CCLS.	767	L4
L3	hop near2 (count\$ or number\$)	1054	L3
L2	(6205146 or 6098091 or 6493318 or 6112239)[pn]	4	L2
L1	6421731[pn]	1	L1

END OF SEARCH HISTORY

WEST Search History

DATE: Thursday, March 27, 2003

Set Name Query

side by side

Hit Count Set Name

result set

DB=USPT; PLUR=YES; OP=ADJ

L12	L11 and l3	8	L12
L11	server near2 farm\$	98	L11
L10	l7 and l9	20	L10
L9	load\$ near2 balanc\$	9995	L9
L8	l3 same l6	5	L8
L7	l3 and l6	45	L7
L6	server near2 (group\$ or farm\$)	1147	L6
L5	l4 and l3	8	L5
L4	(709/105 OR 709/102).CCLS.	767	L4
L3	hop near2 (count\$ or number\$)	1054	L3
L2	(6205146 or 6098091 or 6493318 or 6112239)[pn]	4	L2
L1	6421731[pn]	1	L1

END OF SEARCH HISTORY

WEST Search History

DATE: Thursday, March 27, 2003

Set Name Query

side by side

Hit Count Set Name

result set

DB=USPT; PLUR=YES; OP=ADJ

L14	L13 and l3	22	L14
L13	l9 same request\$ same server\$	325	L13
L12	L11 and l3	8	L12
L11	server near2 farm\$	98	L11
L10	l7 and l9	20	L10
L9	load\$ near2 balanc\$	9995	L9
L8	l3 same l6	5	L8
L7	l3 and l6	45	L7
L6	server near2 (group\$ or farm\$)	1147	L6
L5	l4 and l3	8	L5
L4	(709/105 OR 709/102).CCLS.	767	L4
L3	hop near2 (count\$ or number\$)	1054	L3
L2	(6205146 or 6098091 or 6493318 or 6112239)[pn]	4	L2
L1	6421731[pn]	1	L1

END OF SEARCH HISTORY

WEST Search History

DATE: Thursday, March 27, 2003

Set Name Query
side by side

Hit Count Set Name
result set

DB=USPT; PLUR=YES; OP=ADJ

L17	L16 and l4	7	L17
L16	l13 same (latent\$ or latenc\$ or delay\$)	25	L16
L15	6078943[pn]	1	L15
L14	L13 and l3	22	L14
L13	l9 same request\$ same server\$	325	L13
L12	L11 and l3	8	L12
L11	server near2 farm\$	98	L11
L10	l7 and l9	20	L10
L9	load\$ near2 balanc\$	9995	L9
L8	l3 same l6	5	L8
L7	l3 and l6	45	L7
L6	server near2 (group\$ or farm\$)	1147	L6
L5	l4 and l3	8	L5
L4	(709/105 OR 709/102).CCLS.	767	L4
L3	hop near2 (count\$ or number\$)	1054	L3
L2	(6205146 or 6098091 or 6493318 or 6112239)[pn]	4	L2
L1	6421731[pn]	1	L1

END OF SEARCH HISTORY

Server farm
cluster
group

WEST Search History

DATE: Thursday, March 27, 2003

Set Name Query

side by side

Hit Count Set Name

result set

DB=USPT; PLUR=YES; OP=ADJ

L18	l16 and l11	7	L18
L17	L16 and l4	7	L17
L16	l13 same (latent\$ or latenc\$ or delay\$)	25	L16
L15	6078943[pn]	1	L15
L14	L13 and l3	22	L14
L13	l9 same request\$ same server\$	325	L13
L12	L11 and l3	8	L12
L11	server near2 farm\$	98	L11
L10	l7 and l9	20	L10
L9	load\$ near2 balanc\$	9995	L9
L8	l3 same l6	5	L8
L7	l3 and l6	45	L7
L6	server near2 (group\$ or farm\$)	1147	L6
L5	l4 and l3	8	L5
L4	(709/105 OR 709/102).CCLS.	767	L4
L3	hop near2 (count\$ or number\$)	1054	L3
L2	(6205146 or 6098091 or 6493318 or 6112239)[pn]	4	L2
L1	6421731[pn]	1	L1

END OF SEARCH HISTORY

WEST Search History

DATE: Tuesday, April 01, 2003

Set Name Query
side by side

Hit Count Set Name
result set

DB=EPAB,DWPI; PLUR=YES; OP=ADJ

L2 ((hop or hopping or hopped) near2 (number or count\$)) and (load
near2 (balanc\$ or shar\$))

1 L2

L1 (hop or hopping or hopped) and (load near2 (balanc\$ or shar\$))

9 L1

END OF SEARCH HISTORY

WEST Search History

DATE: Tuesday, April 01, 2003

Set Name Query
side by side

Hit Count Set Name
result set

DB=USPT; PLUR=YES; OP=ADJ

L7	l6 and (udp or icmp)	5	L7
L6	l2 and l5	11	L6
L5	load\$ near2 (shar\$ or balanc\$)	15186	L5
L4	l2 and l3	1	L4
L3	(709/105 OR 709/102).CCLS.	771	L3
L2	(ttl or time-to-live) same (hop or hopping or hopped)	77	L2
L1	(6205146 or 6493318 or 6112239)[pn]	3	L1

END OF SEARCH HISTORY

WEST Search History

DATE: Monday, September 09, 2002

Set Name Query

side by side

Hit Count Set Name

result set

DB=USPT; PLUR=YES; OP=ADJ

L4	L3 and l2	12	L4
L3	l1 near12 (latent\$ or latency or delay\$ or hop\$)	79	L3
L2	(709/105 OR 709/102).CCLS.	715	L2
L1	load near balanc\$	5978	L1

END OF SEARCH HISTORY

WEST

Generate Collection

Print

L4: Entry 1 of 12

File: USPT

Jul 31, 2001

DOCUMENT-IDENTIFIER: US 6269391 B1

TITLE: Multi-processor scheduling kernel

Detailed Description Text (4):

Referring now to FIGS., 1-5, and particularly to FIG. 2, a multi-processing kernel 20 may be embodied to have one or more of four principle features: management of (execution of threads, one-at-a-time) execution exclusion sets by an execution exclusion set module 26; fair-share type scheduling, multi-processing of virtual machines by a multi-processor scheduling module 22; availability of independent scheduling policies for any one, or each, of a plurality of multiple virtual machines; and load balancing using a summed latency of processors, latency of threads, or both to determine when a processor is over or conversely under loaded.

Detailed Description Text (48):

The CPU-specific scheduling module data 64 may also include load balancing data 100. The load balancing data 100 reflects latency of the processor 12 (12a, 12b, or 12c) in question. The latency of a processor 12 reflects, in turn, a summation of the latencies corresponding to all of the threads that have been run during some window of time, on that processor 12 in question. Accordingly, the load balancing data 100 reflects an individual processor's ability to process all threads. By contrast, latency data for an individual thread reflects that individual thread's wait times as experienced across all processors 12 in a multi-processor 10. Accordingly, load balancing data 100 in the CPU-specific scheduling data 64 reflects how heavily loaded a particular processor 12 is. By contrast, latency data within the CPU-specific thread data 64 reflects, or may be interpreted to reflect, the contribution of a particular thread to delays or waits.

Current US Cross Reference Classification (1):709/102

WEST

Generate Collection

Print

L4: Entry 5 of 12

File: USPT

Feb 6, 2001

DOCUMENT-IDENTIFIER: US 6185601 B1

TITLE: Dynamic load balancing of a network of client and server computers

Detailed Description Text (91):

In an alternate embodiment of the invention, load balancing may be initiated, not by the nodes sending a redirect command, but rather by the clients' detection of delays in the processor utilization of the nodes and/or the I/O utilization of the nodes. Each client would maintain a table, listing this utilization, and make decisions similar to those discussed above in connection with FIGS. 7A-D to balance out the load.

Current US Cross Reference Classification (1):709/105

WEST

Generate Collection

Print

L4: Entry 6 of 12

File: USPT

Jan 30, 2001

DOCUMENT-IDENTIFIER: US 6182139 B1

TITLE: Client-side resource-based load-balancing with delayed-resource-binding using TCP state migration to WWW server farmParent Case Text (2):

This application is a continuation-in-part of the co-pending parent application for "A World-Wide-Web Server with Delayed Resource-Binding for Resource-Based Load Balancing on a Distributed-Resource Multi-Node Network", U.S. Ser. No. 08/691,006, filed Aug. 5, 1996, now U.S. Pat. No. 5,774,660.

Detailed Description Text (54):

FIG. 8 highlights multicast from the client-side dispatcher to multiple servers to find the fastest-responding server to handle the request. The invention can also be used in a mode that performs an empirical load-balancing. Connection packets are sent to multiple servers. The first server to respond is likely the server with the lightest load, highest performance, or closest on the network. Thus the first server to respond is often the best choice and would be picked by a rule-based load-balancer. Indeed, such empirical load-balancing may be more efficient than traditional rule-based load balancing since a traditional load balancer's load information is slightly stale due to the latency for load queries.

Current US Cross Reference Classification (1):709/105

WEST



Generate Collection

Print

L4: Entry 8 of 12

File: USPT

Aug 8, 2000

DOCUMENT-IDENTIFIER: US 6101508 A

TITLE: Clustered file management for network resources

Detailed Description Text (99):

In an alternate embodiment of the invention, load balancing may be initiated, not by the nodes sending a redirect command, but rather by the clients' detection of delays in the processor utilization of the nodes and/or the I/O utilization of the nodes. Each client would maintain a table, listing this utilization, and make decisions similar to those discussed above in connection with FIGS. 7A-D to balance out the load.

Current US Cross Reference Classification (3):709/105

WEST Search History

DATE: Monday, September 09, 2002

Set Name Query
side by side

Hit Count Set Name
result set

DB=USPT; PLUR=YES; OP=ADJ

L9	l7 and l2	8	L9
L8	L7[ti,ab]	0	L8
L7	l1 same hop\$	51	L7
L6	l3[ti,ab]	5	L6
L5	6078943[pn]	1	L5
L4	L3 and l2	12	L4
L3	l1 nearl2 (latent\$ or latency or delay\$ or hop\$)	79	L3
L2	(709/105 OR 709/102).CCLS.	715	L2
L1	load near balanc\$	5978	L1

END OF SEARCH HISTORY

WEST

Generate Collection

Print

L9: Entry 1 of 8

File: USPT

Apr 16, 2002

DOCUMENT-IDENTIFIER: US 6374300 B1

TITLE: Method and system for storing load balancing information with an HTTP cookie

Brief Summary Text (15):

In accordance with still other aspects of the present invention, the method provides for employing the server array controller to balance the load demand on the plurality of node servers by determining the optimal node server to receive the HTTP request and generate the HTTP response. The server array controller may employ one of a plurality of functions to determine the optimal node server to balance the load demand. These functions include round trip time, round robin, least connections, packet completion rate, quality of service, server array controller packet rate, topology, global availability, hops, static ratio and dynamic ratio.

Detailed Description Text (24):

Flowing to a block 128, the server array controller 118 makes a load balancing determination and selects the optimal node server to provide access to the requested resource and routes the HTTP request to the selected node server. The server array controller 118 may employ any one of several different types of load balancing methods to analyze metric information and optimally balance client HTTP requests (load demand). These load balancing methods include round trip time, round robin, least connections, packet completion rate, quality of service, server array controller packet rate, topology, global availability, hops, static ratio and dynamic ratio.

Current US Cross Reference Classification (1):709/105

WEST

Generate Collection

Print

L9: Entry 7 of 8

File: USPT

Jul 18, 2000

DOCUMENT-IDENTIFIER: US 6092178 A

TITLE: System for responding to a resource request

Detailed Description Text (27):

In the illustrated non-intrusive mode of operation, status objects 200a, 200b and 200c are invoked on DNS server 100 for the purpose of gathering information from servers 110, 112 and 114, respectively. The configuration and purpose of the status objects may depend upon the load balancing policy that has been selected for choosing a preferred server. For example, where the selected policy requires choosing the least-loaded server or instance of an application (e.g., that which has the fastest response time), each status object may measure the response time of its associated server or the application instance operating on the server. Or, where the selected policy requires choosing the closest server, the status object may be configured to measure the number of hops from DNS server 100 to the object's associated server. Alternatively, each status object may collect all relevant information (rather than a specific piece of information) from its associated server or application instance.

Current US Cross Reference Classification (1):709/105

WEST Search History

DATE: Wednesday, January 08, 2003

<u>Set Name</u>	<u>Query</u>	<u>Hit Count</u>	<u>Set Name</u>
side by side			result set

DB=USPT; PLUR=YES; OP=ADJ

L1	hop near12 (load balanc\$)	12	L1
----	----------------------------	----	----

END OF SEARCH HISTORY

WEST☐ **Generate Collection** **Print**

L1: Entry 1 of 12

File: USPT

Dec 10, 2002

DOCUMENT-IDENTIFIER: US 6493318 B1

TITLE: Cost propagation switch protocols

Brief Summary Text (39):

At each periodic update of cost information, each switch generates a switch cost packet and transmits the packet out on each of its ports in the load balance domain. The switch cost packet includes a hop count field and at least one cost information portion. A switch which receives the switch cost packet updates its cost records in a table entry corresponding to the transmitting switch. As noted herein below, a loop bit offset value may be used as a unique ID to rapidly locate the corresponding table entry. The receiving switch then increments the hop count value in the switch cost packet and transmits the updated packet out on each of its ports within the load balance domain (other than the port from which the packet was just received). The process repeats for each switch receiving the updated switch cost packets from other switches until the process converges by updating all cost information in all switches of the load balance domain.

Detailed Description Text (139):

The hop count field in the first element is set to 0 by the edge switch that initiates the packet and is incremented along the way by each switch the packet encounters. If the hop count gets above 0x0F, it is considered infinite and a path that cannot be taken. This prevents large topologies that may take more than 30 seconds to converge. This does not mean, however, that the topology is limited to 15 switches, but only that a path that takes more than 15 hops is not permitted within a given load balance domain. In effect, the hop count is used to limit the diameter of the network to insure convergence.

Detailed Description Text (141):

As with the other parameters, the retransmission and hop count limits may need to be adjusted as real convergence times are measured. Typically, a load balance domain topology should have a number of short hop routes and not as many long hop routes, since this adds a considerable latency and would defeat some of the benefits of the load balancing. Allowing the advanced user to specify the hop limit within a range may be advantageous, as this could be used to limit the possible number of routes and keep latency at a minimum.

WEST☐ [Generate Collection](#) [Print](#)

L1: Entry 4 of 12

File: USPT

Jul 16, 2002

DOCUMENT-IDENTIFIER: US 6421722 B1

TITLE: Method and apparatus for providing internetworking service reliability

CLAIMS:

3. The method of claim 1, wherein step (c) further comprises determining the new internetworking service based on at least one of exclusion of the at least one flagged internetworking resource, cost of the internetworking and intranetworking resources, minimal number of hops between the internetworking and intranetworking resources, load balancing of the internetworking and intranetworking resources, and bandwidth of the internetworking and intranetworking resources.

18. The multinet network service controller of claim 16, wherein the memory further comprises operational instructions that cause the processing module to determine the new internetworking service based on at least one of exclusion of the at least one flagged internetworking resource, cost of the internetworking and intranetworking resources, minimal number of hops between the internetworking and intranetworking resources, load balancing of the internetworking and intranetworking resources, and bandwidth of the internetworking and intranetworking resources.

WEST Search History

DATE: Wednesday, January 08, 2003

<u>Set Name</u>	<u>Query</u>	<u>Hit Count</u>	<u>Set Name</u>
side by side			result set
<i>DB=USPT; PLUR=YES; OP=ADJ</i>			
L7	l2[ti,ab]	3	L7
L6	l2 and l5	4	L6
L5	(709/105 OR 709/102).CCLS.	753	L5
L4	l2 same (load balanc\$)	3	L4
L3	L2 and (load balanc\$)	21	L3
L2	(hop near2 (number or count)) same server	87	L2
L1	hop near12 (load balanc\$)	12	L1

END OF SEARCH HISTORY

WEST☐ [Generate Collection](#) [Print](#)

L7: Entry 2 of 3

File: USPT

Mar 20, 2001

DOCUMENT-IDENTIFIER: US 6205146 B1

TITLE: Method of dynamically routing to a well known address in a network

Abstract Text (1):

A method of determining an efficient route to a well known address that is particularly applicable to networks that do not have the capability of source routing for calculating routes to specific addresses such as ATM networks based on the IISP protocol. The well known address may represent any entity in the network that provides distributed services (e.g., network server applications) that are to be shared among many nodes and applications on the network, such as LECSs. The method of functions to automatically and dynamically register `well known` addresses on the ports of each node that implements the invention. This permits applications on the network to route to the destination in the shortest possible path thus utilizing network resources in an efficient manner. If there is more than one location with the well known address, e.g., a redundant LECSs in the network, or more than one route to the location then the optimum location will be the one routed to. An optimum location can be determined using any suitable criteria to determine the optimum route to a LECS such as distance, hop count, cost function, link sum, link capacity, etc.

WEST**End of Result Set**

Generate Collection

Print

L4: Entry 3 of 3

File: USPT

Jul 18, 2000

DOCUMENT-IDENTIFIER: US 6092178 A

TITLE: System for responding to a resource request

Detailed Description Text (27):

In the illustrated non-intrusive mode of operation, status objects 200a, 200b and 200c are invoked on DNS server 100 for the purpose of gathering information from servers 110, 112 and 114, respectively. The configuration and purpose of the status objects may depend upon the load balancing policy that has been selected for choosing a preferred server. For example, where the selected policy requires choosing the least-loaded server or instance of an application (e.g., that which has the fastest response time), each status object may measure the response time of its associated server or the application instance operating on the server. Or, where the selected policy requires choosing the closest server, the status object may be configured to measure the number of hops from DNS server 100 to the object's associated server. Alternatively, each status object may collect all relevant information (rather than a specific piece of information) from its associated server or application instance.

WEST☐ Generate Collection

Print

L4: Entry 1 of 3

File: USPT

Nov 27, 2001

DOCUMENT-IDENTIFIER: US 6324580 B1

TITLE: Load balancing for replicated services

Brief Summary Text (14):

Depending upon the selected policy, status objects or modules are created to collect information from each server offering the replicated service or application that is being load-balanced. The information collected from each server may include the number of requests held and/or processed by the server or service, the response time and/or operational status (e.g., is it up or down) of the server or service, the distance (e.g., the number of network hops) to the server, etc.

WEST Search History

DATE: Wednesday, January 08, 2003

<u>Set Name</u>	<u>Query</u>	<u>Hit Count</u>	<u>Set Name</u>
side by side			result set
<i>DB=USPT; PLUR=YES; OP=ADJ</i>			
L9	l7 and l5	8	L9
L8	l6 and l7	0	L8
L7	(709/105 OR 709/102).CCLS.	753	L7
L6	L5[ti,ab]	46	L6
L5	hop near2 (number or count\$)	1007	L5
L4	6205146[pn]	1	L4
L3	6205146[uref]	2	L3
L2	6249801[uref]	4	L2
L1	6078943[uref]	10	L1

END OF SEARCH HISTORY

WEST☐ **Generate Collection** **Print**

L9: Entry 6 of 8

File: USPT

Aug 1, 2000

DOCUMENT-IDENTIFIER: US 6098091 A

TITLE: Method and system including a central computer that assigns tasks to idle workstations using availability schedules and computational capabilities

Detailed Description Text (13):

Referring now to FIG. 3B, scheduler 304 coordinates the tasks file 312 with the resource available file 308. The information in resource available file 308 of FIG. 3B was transmitted by the remote computers to the central computer. Each remote computer may be associated with a machine identification (e.g. Internet address), a computer serial number or a phone number at which the remote computer can be reached. This information is indexed in a computer identifier field 332. Corresponding with the computer identifier field 332 may be other data fields such as a time available field 336 indicating when the remote computer is available. The time when the remote computer is available should be given in a worldwide standardized time frame, such as the time of day in Greenwich mean time. Other possible corresponding data fields include the benchmark rating field 338 indicating hardware capabilities available at the remote computer, the processor field 340 indicating the type of processor used by the remote computer, the RAM field 342 indicating RAM available on the remote computer, the memory field 344 including memory available for long term storage 344, the number of hops to an Internet backbone field 345 (indicating the number of hops which is the number of routers or switches between the backbone and the remote computer), the transmission bandwidth field 346 indicating the transmission bandwidth of communications with the remote computer and other data fields 347 which may indicate other hardware such as processing cards that may be available. All of this information may be coded so that the central computer can appropriately assign a task to be completed to remote computers most efficiently able to complete the task. At the times when the remote computer has agreed to operate in a contractor relationship, the central computer using the management program will follow scheduler instructions and will send the assigned task to the remote computer.

Current US Cross Reference Classification (2):
709/102

CLAIMS:

8. The system of claim 1 wherein said computational capabilities of said remote computer include:

type of processor used in said remote computer;

benchmark rating of computing power of said remote computer;

amount of random-access-memory (RAM) available in said remote computer;

number of hops between an Internet backbone and said remote computer; and

transmission bandwidth of said remote computer.

15. The management apparatus of claim 14 wherein said computational capabilities of said remote computer include:

type of processor used in said remote computer;

benchmark rating of computing power of said remote computer;

amount of random-access-memory (RAM) available in said remote computer;

number of hops between an Internet backbone and said remote computer; and

transmission bandwidth of said remote computer.

WEST☐ **Generate Collection** **Print**

L9: Entry 5 of 8

File: USPT

Aug 29, 2000

DOCUMENT-IDENTIFIER: US 6112239 A

TITLE: System and method for server-side optimization of data delivery on a distributed computer networkDetailed Description Text (7):

Several delivery, or "mirror" sites are shown connected to the Internet 10 in FIG. 1. A first delivery site 26 might be located a small number of "hops" from the first user terminal 12. A second delivery site 28 might be located further away from the first user terminal 12, but close to the third user terminal 20. A third delivery site 30 might be as close to the third user terminal 20 as the second delivery site 28 is. As previously noted, a user and a provider or delivery site that are "geographically" near each other might not be "electronically" near each other on the Internet. By decreasing the "electronic" distance between the user and the provider or delivery site, the number of network connections and routers over which data must travel can be decreased.

Current US Cross Reference Classification (1):709/105

WEST☐ **Generate Collection** **Print**

L9: Entry 4 of 8

File: USPT

Aug 29, 2000

DOCUMENT-IDENTIFIER: US 6112248 A

TITLE: Method and system for dynamically balancing network traffic using address resolution protocol

Brief Summary Text (15):

The basic algorithm of the OSPF protocol is shown in FIG. 4. The data processing device broadcasts a message including the networks it can reach and the distances to these networks determined by the number of hops (step 822), and also receives such messages from other data processing devices (step 823). When a route changes (step 824), each router calculates the shortest path from itself to each of the networks (step 825) and sets its routing table according to the paths (step 826).

Detailed Description Text (6):

FIG. 6 shows an example 73 of a global routing table for the network 2, i.e., the network in question. For each target network 731, this table shows a router (ID) 732 to this network. For each router, this table shows a physical address 733 of the interface to the target network, a network address 734 in the network in question, the next hop 735 to the target network after this router, and a priority 736 of the route. This priority is set by the network administrator according to the number of hops up to the target network and according to the network processing capability. Hence, the best route will be selected preferentially. In FIG. 6, priority is 1 when the number of hop is 1 and priority is 2 when the number of hops are 2.

Current US Cross Reference Classification (2):709/105

WEST Search History

DATE: Wednesday, January 08, 2003

<u>Set Name</u>	<u>Query</u>	<u>Hit Count</u>	<u>Set Name</u>
side by side			result set
<i>DB=USPT; PLUR=YES; OP=ADJ</i>			
L11	l10 and l7	0	L11
L10	6098091[uref]	4	L10
L9	l7 and l5	8	L9
L8	l6 and l7	0	L8
L7	(709/105 OR 709/102).CCLS.	753	L7
L6	L5[ti,ab]	46	L6
L5	hop near2 (number or count\$)	1007	L5
L4	6205146[pn]	1	L4
L3	6205146[uref]	2	L3
L2	6249801[uref]	4	L2
L1	6078943[uref]	10	L1

END OF SEARCH HISTORY

WEST Search History

DATE: Wednesday, January 08, 2003

<u>Set Name</u>	<u>Query</u>	<u>Hit Count</u>	<u>Set Name</u>
side by side			result set
<i>DB=USPT; PLUR=YES; OP=ADJ</i>			
L15	(server near2 farm) and l5	8	L15
L14	6078943[pn]	1	L14
L13	6249801[pn]	1	L13
L12	6493318[pn]	1	L12
L11	l10 and l7	0	L11
L10	6098091[uref]	4	L10
L9	l7 and l5	8	L9
L8	l6 and l7	0	L8
L7	(709/105 OR 709/102).CCLS.	753	L7
L6	L5[ti,ab]	46	L6
L5	hop near2 (number or count\$)	1007	L5
L4	6205146[pn]	1	L4
L3	6205146[uref]	2	L3
L2	6249801[uref]	4	L2
L1	6078943[uref]	10	L1

END OF SEARCH HISTORY

WEST**End of Result Set**☐ **Generate Collection** **Print**

L15: Entry 8 of 8

File: USPT

Jun 20, 2000

DOCUMENT-IDENTIFIER: US 6078953 A

TITLE: System and method for monitoring quality of service over network

Detailed Description Text (3):

Network QoS occurs by managing the resources that serve network application traffic, for example. This typically includes the following resources: link bandwidth, application server bandwidth (CPU), and buffer space on generally all nodes (end-points, routers and gateways). Typically, data through-put is limited by the speed of Internet access links and by the server CPU capacity, and response time is determined by the number of hops in a route, physical length of the route, and extent of congestion in the route. There are various other factors that may affect QoS, such as the behavior of TCP/IP, severe congestion anywhere in the route, prioritization of traffic along the route, etc. To a network administrator, embodiments of the present invention provide discrimination of different traffic types and provide methods for enforcement of traffic flow by management to the above resources.

Detailed Description Text (30):

The present invention takes into account, in one or more embodiments, the factors which are described specifically above. Although the above has been generally described in terms of a specific type of information, other types of information on a network can also be used with the present invention. Additionally, the present invention has been described in general to a specific system. For instance, the present bandwidth management tool can be applied at a network's Internet access link. Alternatively, the present tool can be applied to a private WAN link to a remote corporate site or an access to a server farm (e.g., a group of servers located in a special part of the network close to an access link, e.g., in a web hosting environment). Alternatively, the present invention can be applied to key servers (e.g., database/web server) within an organization servicing internal and/or external users. Furthermore, the present bandwidth management tool can be applied to any combination of the above or the like.